



**IMPLEMENTASI SELF BALANCING ROBOT DENGAN  
MIKROKONTROLER ARDUINO DAN SENSOR  
AKSELEROMETER**

Disusun dan Diajarkan untuk Memenuhi Persyaratan Ujian Akhir Memperoleh  
Gelar Sarjana Komputer pada Fakultas Sains dan Teknologi  
Universitas Pembangunan Panca Budi  
Medan

---

**SKRIPSI**

---

**OLEH**

**NAMA : MISWAN DANI**  
**NIM : 1514370007**  
**PROGRAM STUDI : SISTEM KOMPUTER**

**FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS PEMBANGUNAN PANCA BUDI  
MEDAN  
2019**

# **ABSTRAK**

**MISWAN DANI**

Program Studi Sistem Komputer, Fakultas Sains dan Teknologi  
Universitas Pembangunan Panca Budi Medan  
E-Mail : miswand975@gmail.com

## **Self Balancing Robot Dengan Mikrokontroler Arduino dan Sensor Accelerometer 2019**

Self Balancing Robot adalah salah satu penerapan sistem pendulum terbalik dimana sistem tersebut memiliki gravitasi yang berada diatas poros putar sehingga menyebabkan sistem kurang stabil dan tidak seimbang. Untuk itu, diperlukan suatu sistem kendali yang dapat membuat pendulum seimbang dengan menggerakkan dua buah roda yang menjadi poros putar dari pendulum terbalik. Tujuan dari penelitian ini adalah merancang suatu sistem pengendalian robot beroda dua agar dapat mempertahankan posisi dalam keadaan seimbang. Sistem ini mendapatkan masukan dari sensor MPU-6050 yaitu gabungan antara akselerometer dan giroskop. Sistem pengendalian yang digunakan yaitu metode kendali yang mengacu pada titik keseimbangan yang dihasilkan dari sensro MPU – 6050. Nantinya titik keseimbangan akan dimasukkan ke arduino nano untuk mengontrol arah putar Motor DC agar dapat bekerja menyeimbangkan posisi robot.

**Kata Kunci** : Self Balancing Robot, MPU – 6050, arduino nano

## KATA PENGANTAR

Puji dan syukur kita panjatkan kehadirat Tuhan Yang Maha Esa yang telah memberikan rahmat dan hidayah-Nya kepada kita semua sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Skripsi ini merupakan salah satu syarat yang harus dipenuhi untuk menyelesaikan pendidikan S-I pada Program Studi Sistem Komputer Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.

Dalam menyelesaikan skripsi ini penulis banyak mengalami kesulitan namun berkat bimbingan, dorongan, dan bantuan dari rekan-rekan mahasiswa – mahasiswi Program Studi Sistem Komputer Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan, dosen dan semua pihak yang terkait dalam penyelesaian skripsi ini, hal tersebut dapat penulis atasi dengan baik. Selama pengerjaan skripsi ini penulis banyak mendapat bantuan, dorongan, motivasi baik secara langsung maupun tidak langsung. Maka pada kesempatan ini, penulis mengucapkan terima kasih kepada :

1. **Orang Tua** yang telah memberikan doa dan dukungannya kepada penulis.
2. **Bapak Dr. H. Muhammad Isa Indrawan, SE., MM.**, selaku Rektor Universitas Pembangunan Panca Budi Medan.

3. **Ibu Sri Shindi Indira, ST., M.Sc** selaku Dekan Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
4. **Bapak Muhammad Iqbal, S.Kom., M.Kom.**, selaku Ketua Program Studi Sistem Komputer Fakultas Ilmu Komputer Universitas Pembangunan Panca Budi Medan.
5. **Bapak Solly Aryza, S.T., M.Eng** selaku Dosen pembimbing I yang telah membimbing dan mengarahkan penulis dalam menyelesaikan Skripsi ini.
6. **Bapak Amani Darma Tarigan, S.T., M.T** selaku dosen pembimbing II yang telah membimbing dan mengarahkan penulis dalam menyelesaikan Skripsi ini.
7. **Sahabat Hartina Danty, Uci, Irvan** yang telah memberikan doa dan dukungan

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat banyak kekurangan dan kesalahan. Oleh karena itu, penulis sangat mengharapkan saran dan kritik untuk perbaikan penulisan di kemudian hari. Akhir kata, semoga skripsi ini dapat memberi manfaat dan menambah wawasan maupun pengetahuan kita

Medan, 19 Agustus 2019

Penulis,

**(MISWAN DANI)**

**NPM : 1514370007**

## DAFTAR ISI

<b>LEMBAR JUDUL</b>	
<b>LEMBAR PENGESAHAN</b>	
<b>KATA PENGANTAR</b> .....	i
<b>DAFTAR ISI</b> .....	iii
<b>DAFTAR TABEL</b> .....	vi
<b>DAFTAR GAMBAR</b> .....	vii
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
<b>BAB II LANDASAN TEORI</b>	
2.1 Definisi Robot.....	4
2.1.1 Sejarah dan Perkembangan Robot .....	4
2.1.2 Robot Keseimbangan ( <i>self balancing robot</i> ) .....	7
2.2 Sensor Accelerometer Modul MPU 6050 .....	9
2.3 Definisi Arduino .....	10
2.3.1 Sejarah Arduino .....	11

2.3.2 Software Arduio.....	13
2.3.3 Arduino Nano.....	14
2.4 Motor Steper .....	19
2.4.1 Prinsip Kerja Motor Steper .....	20
2.4.2 Karakteristik Motor Steper .....	21
2.4.3 Jenis – Jenis Motor Steper .....	22
2.5 UML ( <i>Unified Modeling Language</i> ).....	27
1. Use Case Diagram .....	27
2. Activity Diagram .....	29
3. Class Diagram.....	30
4. Squance Diagram.....	32
<b>BAB III METODE PENELITIAN</b>	
3.1 Tahapan Penelitian.....	35
3.2 Metode Pengumpulan Data.....	36
1. Penelitian Kepustakaan ( <i>Library Research</i> ).....	36
2. Penelitian Labororium .....	36
3.3 Analisis Sistem yang Sedang Berjalan.....	37
1. Analisis Sistem yang Berjalan .....	37
2. Analisis Kebutuhan Sistem.....	38
3. Analisis Pemodelan Sistem .....	38
3.4 Rancangan Penelitian .....	47
1. Perangkat Keras .....	47

2. Perangkat Lunak .....	51
--------------------------	----

#### **BAB IV HASIL DAN PEMBAHASAN**

4.1 Kebutuhan Spesifikasi Minimum Hardware dan Software .....	54
4.2 Pengujian Aplikasi dan Pembahasan .....	54
1. Pengujian Arduino .....	55
2. Pengujian Sensor <i>Accelerometer</i> MPU 6050 .....	55
3. Pengujian Keseimbangan Robot.....	59
4. Pengujian Robot Secara Keseluruhan.....	63

#### **BAB V PENUTUP**

5.1 Simpulan .....	65
5.2 Saran .....	65

#### **DAFTAR PUSTAKA**

#### **LAMPIRAN – LAMPIRAN**

## DAFTAR TABEL

	HALAMAN
Tabel 2.1 Simbol – simbol <i>Use Case</i> Diagram .....	28
Tabel 2.2 Simbol – simbol <i>Activity</i> Diagram .....	29
Tabel 2.3 Simbol – simbol <i>Class</i> Diagram .....	31
Tabel 2.4 Simbol – simbol <i>Squence</i> Diagram.....	32
Tabel 3.1 Deskripsi Aktor .....	39
Tabel 3.2 ID <i>Uce Case</i> .....	39
Tabel 3.3 Use Case Mengaktifkan Robot .....	40
Tabel 3.4 Use Case Menggerakkan Robot Maju dan Mundur.....	41
Tabel 3.5 Use Case Memberikan Beban dan Gangguan pada Robot .....	42
Tabel 4.1 Data Pengujian Robot Menyeimbangkan diri .....	59
Tabel 4.2 Data Pengujian Keseimbangan Robot.....	60
Tabel 4.3 Data Pengujian Keseimbangan Robot Diberi gangguan .....	61
Tabel 4.4 Data Pengujian Keseimbangan Robot Diberi beban .....	62
Tabel 4.5 Data Pengujian Keseimbangan Robotdiberi beban, gangguan .....	62
Tabel 4.6 Data Pengujian Keseluruhan Sistem .....	63



## DAFTAR GAMBAR

	HALAMAN
Gambar 2.1 <i>Balancing Robot</i> Beroda dua Menyeimbangkan diri .....	8
Gambar 2.2 Sensor <i>Accelerometer</i> .....	9
Gambar 2.3 <i>Toolbar</i> Arduino.....	13
Gambar 2.4 Arduino Nano .....	15
Gambar 2.5 Motor Stepper.....	19
Gambar 2.6 Prinsip Kerja Motor Stepper .....	21
Gambar 2.7 Motor stepper tipe Variable Reluctance .....	23
Gambar 2.8 Motor stepper tipe Permanent Magnet .....	24
Gambar 2.9 Motor stepper tipe Hybrid .....	25
Gambar 2.10 Motor Stepper dengan lilitan Unipolar.....	26
Gambar 2.11 Motor Stepper dengan lilitan Bipolar .....	27
Gambar 3.1 <i>Flowchart</i> Tahapan Penelitian .....	35
Gambar 3.2 <i>Use Case</i> Diagram Aktor .....	40
Gambar 3.3 <i>Activity</i> Diagram Mengaktifkan Robot .....	43
Gambar 3.4 <i>Activity</i> Diagram Menggerakkan Robot Maju dan Mundur.	44
Gambar 3.5 <i>Activity</i> Diagram Memberikan Beban dan Gangguan.....	45
Gambar 3.6 <i>Squence</i> Diagram Mengaktifkan Robot .....	45
Gambar 3.7 <i>Squence</i> Diagram Menggerakkan RobotMaju dan mundur	46
Gambar 3.8 <i>Squence</i> Diagram Memberikan Beban dan gangguan .....	47
Gambar 3.9 Diagram <i>Block</i> Perancangan <i>Self Balancing Robot</i> .....	47

Gambar 3.10 Rangkaian Mikrokontroler Arduino Nano .....	48
Gambar 3.11 Rangkaian <i>MPU 6050</i> .....	49
Gambar 3.12 Rangkaian <i>Stepper Motor Driver Carrier</i> .....	50
Gambar 3.13 Rangkaian keseluruhan sistem. ....	50
Gambar 3.14 <i>Flowchart</i> Keseluruhan Sistem. ....	51
Gambar 3.15 <i>Flowchart</i> Sensor <i>Accelerometer</i> .....	52

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi robotika pada masa sekarang ini sangatlah pesat, pada dunia industri robot sangat membantu meningkatkan kualitas dan kuantitas dari pada hasil industry. Teknologi robotika juga sudah merambah sampai dunia pendidikan dan dunia hiburan. Untuk menambah fungsi dan kecerdasan pada teknologi robot dapat menambahkan sensor dan control. Sebagai contoh adalah *self balancing robot*.

*Self balancing robot* (robot penyeimbang) adalah sebuah robot yang memiliki dua buah roda disisi samping kanan dan kirinya, dan menggunakan kontroler untuk penyeimbang. Balancing robot merupakan pengembangan dari model pendulum terbalik (*inverted pendulum*) yang diletakkan diatas dua buah roda. Untuk mempertahankan posisi seimbang tegak lurus dari robot diperlukan kontrol dan rangkaian hardware yang baik. Dan *segwai* adalah sebuah alat transportasi yang menggunakan *self balancing robot* dan sensor penyeimbang untuk kendali yang akan membuatnya berjalan, berbelok ke kanan dan ke kiri juga berhenti.

Pada penelitian ini adalah untuk merancang *self balancing robot* yang dapat mempertahankan keseimbangannya pada permukaan bumi yang berbidang datar. Pada penelitian ini menggunakan beberapa alat seperti, sensor modul

MPU 6050, *module bluetooth hc 05*, Arduino nano dan motor stepper sebagai penggerak roda self balancing robot. Dari penjelasan diatas penulis mengambil judul “**Implementasi Self Balancing Robot dengan Mikrikontroler Arduino dan Sensor Accelerometer**”

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang diatas maka dapat diuraikan rumusan masalah pada penelitian ini adalah :

1. Bagaimana cara membuat sistem penggerak dalam robot ?
2. Bagaimana merancang *self balancing robot* yang dapat berdiri tegak seimbang pada permukaan bumi ?
3. Bagaimana mengaplikasikan sensor modul MPU 6050 pada robot keseimbangan ?

## **1.3 Batasan Masalah**

Berikut ini beberapa batasan masalah yang akan dibahas pada penelitian ini adalah :

1. Mikrokontroler yang digunakan adalah arduino nano
2. Robot diprogram dengan menggunakan *software* arduino IDE
3. Robot hanya berdiri seimbang pada permukaan bidang datar
4. Menggunakan suplai daya battery 9V
5. Robot ini tidak mengikuti jalur atau garis tertentu

#### **1.4 Tujuan Penelitian**

Berikut adalah tujuan dari pembuatan tugas akhir ini adalah sebagai berikut :

1. Merancang sistem penggerak pada robot.
2. Merancang robot yang dapat menyeimbangkan posisi tegak lurus terhadap permukaan bumi pada bidang datar dengan menggunakan sensor Akselerometer dan Arduino Nano.
3. Mengaplikasikan sensor modul MPU 6050 terhadap robot keseimbangan sebagai sensor penyeimbang robot.

#### **1.5 Manfaat**

Berikut ini beberapa manfaat dari penelitian yang akan dibahas sebagai berikut :

1. Mengetahui pengaplikasian sensor modul MPU 6050 terhadap robot keseimbangan.
2. Mengetahui cara kerja robot keseimbangan pada permukaan bumi pada bidang datar.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Definisi Robot**

Robot berasal dari kata “*robota*” yang dalam bahasa Ceko yang berarti budak, pekerja atau kuli. Pertama kali kata “*robota*” diperkenalkan oleh Karel Capek dalam sebuah pentas sandiwara pada tahun 1921 yang berjudul RUR (Rossum’s Universal Robot). Pentas ini mengisahkan mesin yang menyerupai manusia yang dapat bekerja tanpa lelah yang kemudian memberontak dan menguasai manusia. Istilah “robot” ini kemudian mulai terkenal dan digunakan untuk menggantikan istilah yang dikenal saat itu yaitu automation. Dari berbagai literatur robot dapat didefinisikan sebagai sebuah alat mekanik yang dapat diprogram berdasarkan informasi dari lingkungan (melalui sensor) sehingga dapat melaksanakan beberapa tugas tertentu baik secara otomatis ataupun tidak sesuai program yang di inputkan berdasarkan logika. (Ii & Pustaka, 2013)

##### **2.1.1 Sejarah dan Perkembangan Robot**

Robot pada awalnya diciptakan untuk menggantikan kerja manusia untuk sesuatu yang berulang, membutuhkan ketepatan yang tinggi dan juga untuk menggantikan manusia bila harus berhubungan dengan daerah berbahaya. Penggunaan robot lebih banyak terletak pada industri, misalnya untuk proses welding pada industri otomotif. Selain pada industri, penggunaan robot semakin

berkembang luas. Sementara itu, pada dunia pendidikan di tingkat universitas telah dilakukan berbagai macam kontes yang memacu para akademisi dan mahasiswa dalam melakukan riset tentang robot. Kedepannya, robot akan semakin berkembang sehingga mampu bergerak dan berpikir seperti manusia berdasarkan logika-logika pemograman yang diinputkan.

Seiring berkembangnya teknologi, berbagai robot dibuat dengan spesialisasi atau keistimewaan tertentu. Robot dengan keistimewaan tertentu sangat erat kaitannya dengan pemenuhan kebutuhan dalam dunia industri modern, dimana industri modern menuntut adanya suatu alat dengan kemampuan tinggi yang dapat membantu menyelesaikan pekerjaan manusia ataupun menyelesaikan pekerjaan yang tidak mampu diselesaikan manusia. Pemanfaatan teknologi robot mempunyai sisi lain yang mendatangkan ancaman bagi sebagian orang, karena kehilangan kesempatan kerja. Dari survei yang dilakukan terhadap pemakai robot di Inggris, penghematan tenaga kerja ditulis sebagai faktor terpenting dalam mengambil keputusan untuk mengadopsi robot. Meskipun demikian, walau beberapa pekerjaan dan tugas dihasilkan dengan campur tangan robot, tetapi terdapat kecenderungan untuk tidak menggantikan tenaga manusia seluruhnya. Secara teoritis robot dimasukkan bukan pada faktor produksi yang berupa masukan buruh, melainkan pada masukan modal.

Negara yang paling getol mengadakan penelitian mengenai berbagai macam robot ini adalah Jepang. Hal ini tak lain karena Jepang juga gigih dalam melakukan penelitian teknologi infrastruktur seperti komponen dan piranti mikro (microdevices) yang akhirnya bidang ini terbukti sebagai inti dari pengembangan

robot modern. Sebenarnya, robot bukanlah 'barang baru' bagi masyarakat Jepang. Robot pertama Jepang sudah diciptakan berabad-abad yang lalu. Tentunya tidak dengan bentuk yang ada saat ini. Mulai dari robot yang bisa menyirami sawah buatan Kaya no Miko seperti yang diceritakan dalam koleksi cerita abad ke-12, Konjaku Monogatari Shu, hingga boneka robot karakuri-ningyo yang dikembangkan dengan tingkat teknologi yang cukup tinggi dan ditampilkan dalam bentuk boneka sebagai hiburan di teater dan dalam festival (hingga sekarang tetap ditampilkan dalam Festival Takayama di Prefektur Gifu).

Pada tahun 1927 muncul robot Jepang yang pertama yang dikembangkan dengan mempergunakan teknologi barat, diberi nama Gakutensoku. Robot ini bisa tersenyum, mengedip-ngedipkan mata dan bahkan bisa menulis. Dengan adanya pengembangan robot ini, robot kini bisa menjadi teman, mempunyai kecerdasan, dan perasaan manusia, seperti dalam cerita kartun Astro Boy.

Keunggulan dalam teknologi robot tak dapat dipungkiri, telah lama dijadikan ikon kebanggaan negara-negara maju di dunia. Kecanggihan teknologi yang dimiliki, gedung-gedung tinggi yang mencakar langit, tingkat kesejahteraan rakyatnya yang tinggi, kota-kotanya yang modern, belumlah terasa lengkap tanpa popularitas kepiawaian dalam dunia robot. Pada awalnya, aplikasi robot hampir tak dapat dipisahkan dengan industri sehingga muncul istilah industrial robot dan robot manipulator. Definisi yang populer ketika itu, robot industri adalah suatu robot tangan (arm robot) yang diciptakan untuk berbagai keperluan dalam meningkatkan produksi, memiliki bentuk lengan-lengan kaku yang terhubung secara seri dan memiliki sendi yang dapat bergerak berputar (rotasi) atau



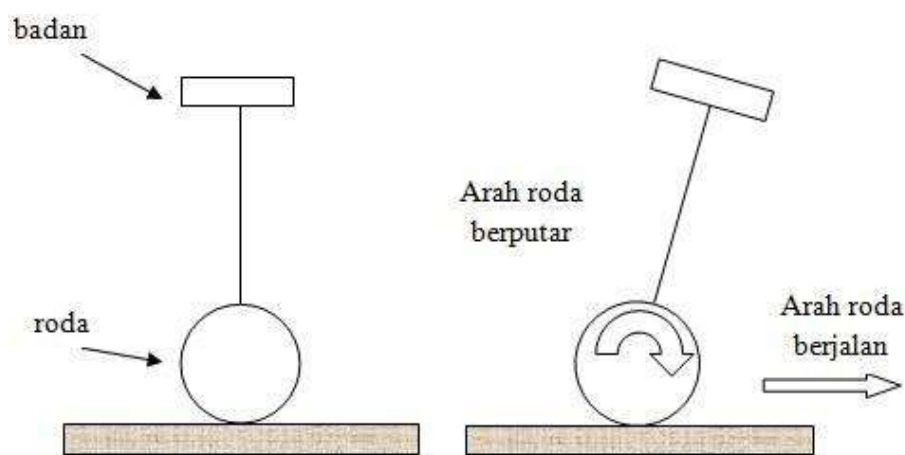
memanjang/memendek (translasi atau prismatic). Satu sisi lengan yang disebut sebagai pangkal ditanam pada bidang atau meja yang statis (tidak bergerak), sedangkan sisi yang lain yang disebut sebagai ujung (*end effector*) dapat dimuati dengan tool tertentu sesuai dengan tugas robot. Dalam dunia mekanikal, manipulator ini memiliki dua bagian, yaitu tangan atau lengan (*arm*) dan pergelangan (*wrist*). Pada pergelangan ini dapat diinstall berbagai tool. Begitu diminatinya penggunaan manipulator dalam industry, menyebabkan banyak perusahaan besar di dunia menjadikan robot industri sebagai unggulan. Bahkan beberapa perusahaan di Jepang masih menjadikan manipulator sebagai produk utamanya, seperti *Fanuc Inc.* yang memiliki pabrik utamanya di lereng gunung Fuji.

Dewasa ini mungkin definisi robot industri itu sudah tidak sesuai lagi karena teknologi mobile robot juga sudah dipakai meluas sejak awal tahun 1980-an. Seiring itu pula kemudian muncul istilah humanoid robot (konstruksi mirip manusia), animaloid (mirip binatang), dan sebagainya. Bahkan kini dalam industri spesifik seperti industri perfilman, industri angkasa luar dan industri pertahanan atau mesin perang, arm robot atau manipulator bisa jadi hanya menjadi bagian saja sistem robot secara keseluruhan.

### **2.1.2 Robot Keseimbangan (*Self Balancing Robot*)**

Pendulum terbalik adalah pendulum yang terengsel ke kereta beroda yang dapat bergerak maju dan mundur pada bidang horisontal di sepanjang lintasan. Penerapan konsep pendulum terbalik dalam dunia robotika bisa dilihat

pada robot keseimbangan, yaitu robot dengan dua roda yang roda tersebut diasumsikan sebagai kereta beroda dan badan robot diasumsikan sebagai pendulum. Sistem ini tidak stabil karena ketika kereta beroda diberi gangguan dari luar maka pendulum akan jatuh. Ketika pendulum atau balancing robot, mempertahankan agar pendulum tidak terjatuh dibutuhkan sebuah kendali suatu kendali khusus.(BOBBY, SUSANTO, & SURATMAN, 2017)



**Gambar 2.1 Balancing Robot Berdoa dua Menyeimbangkan diri**

(Pamungkas, Elektro, & Nuswantoro, 2014)

Dalam keadaan diam, pendulum terbalik yang diatur agar pada keadaan awal yang tegak, akan mulai membentuk sudut  $\theta$  dan lama kelamaan akan jatuh karena adanya gaya gravitasi. Untuk mempertahankan posisi pendulum pada suatu titik, dipertahankan diperlukan sebuah gaya yang dapat menahan pergerakan pendulum. Cara menghasilkan gaya tersebut adalah dengan membuat kereta tersebut maju kearah dengan arah kemana pendulum tersebut condong / akan jatuh. Kendali yang baik akan membuat pendulum tetap seimbang dengan cara mengendalikan pergerakan poros putar atau kereta beroda

dimana pendulum tersebut terpasang. Dasar untuk membuat robot beroda dua dapat seimbang adalah dengan cara mengendalikan roda searah dengan arah jatuhnya bagian atas robot tersebut. Apabila proses tersebut dapat terlaksana maka robot tersebut dapat seimbang.

## 2.2 Sensor *Accelerometer* Modul MPU 6050

Sensor *Accelerometer* yaitu sebuah transduser yang bekerja untuk mengukur dan mendeteksi adanya percepatan dan getaran akibat gravitasi bumi. Alat sensor ini mampu bekerja untuk mengukur suatu getaran yang terjadi pada benda atau objek tertentu seperti mesin, kendaraan, bangunan, dan lebih hebatnya lagi bisa mengukur getaran di dalam bumi tanpa pengaruh gravitasi bumi.



**Gambar 2.2 Sensor *Accelerometer***

(Sumber : Grace Bobby dkk, 2015)

Berdasarkan ilmu Fisika dan Hukum Fisika bahwa bila konduktor digerakan ke medan magnet ataupun sebaliknya, maka yang terjadi adalah akan menimbulkan suatu tegangan induksi pada konduktor yang ada. Sensor *Accelerometer* yang diletakan di permukaan bumi itu bisa mendeteksi suatu

percepatan 1g atau sebuah ukuran gravitasi pada bumi di titik pusat vertical. Untuk pendeteksi bumi pada titik *vertical*, karena disebabkan adanya pergerakan percepatan dari *horisontal*, oleh karena itu Sensor *Accelerometer* ini akan mengukur percepatannya pada waktu itu atau secara langsung ketika percepatan itu bergerak secara horisontal. Karena Sensor *Accelerometer* ini dibuat oleh perusahaan atau pabriknya yang dibuat secara berbeda mulai dari jenis dan spesifikasinya karena alat ini sangat khusus untuk setiap digunakannya. Sekarang pada saat ini, Hampir semua alat ini berpenampilan Digital.

*Accelerometer* MPU-6050 adalah sebuah modul berinti MPU-6050 yang merupakan 6 *axis Motion Processing Unit* dengan penambahan regulator tegangan dan beberapa komponen pelengkap lainnya yang membuat modul ini siap pakai dengan tegangan *supply* sebesar 3-5 VDC. Modul ini memiliki *interface* I2C yang dapat disambungkan langsung ke MCU (*Microcontroller Units*) yang memiliki fasilitas I2C. Sensor MPU-6050 berisi sebuah *Microelectromechanical Systems* (MEMS) *Accelerometer* dan sebuah MEMS Gyro yang saling terintegrasi. Sensor ini sangat akurat dengan fasilitas hardware internal 16 bit ADC untuk tiap kanalanya. Sensor akan menangkap nilai kanal axis X, Y, dan Z bersamaan dalam satu waktu.

### **2.3 Definisi Arduino**

Arduino adalah sistem punarupa elektronika (*electronic prototyping platform*) berbasis *open-source* yang fleksibel dan mudah digunakan baik dari sisi perangkat keras/*hardware* maupun perangkat lunak/*software*. Di luar itu,

kekuatan utama arduino adalah jumlah pemakai yang sangat banyak sehingga tersedia pustaka kode program (*code library*) maupun modul pendukung (*hardware support modules*) dalam jumlah yang sangat banyak. Hal ini memudahkan para pemula untuk mengenal dunia mikrokontroler.

Arduino didefinisikan sebagai sebuah platform elektronik yang *open source*, berbasis pada *software* dan hardware yang fleksibel dan mudah digunakan, yang ditujukan untuk seniman, desainer, hobbies dan setiap orang yang tertarik dalam membuat sebuah objek atau lingkungan yang interaktif.(Artanto, 2012)

Arduino sebagai sebuah *platform* komputasi fisik (*Physical Computing*) yang *open source* pada *board input output* sederhana, yang dimaksud dengan *platform* komputasi fisik disini adalah sebuah sistem fisik yang interaktif dengan penggunaan *software* dan hardware yang dapat mendeteksi dan merespon situasi dan kondisi.

### **2.3.1 Sejarah Arduino**

Modul *hardware* Arduino diciptakan pertama kali di Ivrea, Italia pada tahun 2005 oleh Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David A. Mellis, dan Nicholas Zambetti (Arduino, 2011 dan Banzi 2008).

Bahasa Arduino merupakan *fork* (turunan) bahasa *Wiring Platform* dan bahasa *Processing*. *Wiring Platform* diciptakan oleh Hernando Barragan pada tahun 2003 (Wiring.org.co, 2011) dan *Processing* dibuat oleh Casey Reas dan Benjamin Fry pada tahun 2011 (*Processing.org*, 2012).

Arduino dikembangkan dari thesis Hernando Barragan di desain interaksi Institute Ivrea. Arduino dapat menerima masukan dari berbagai macam sensor dan juga dapat mengontrol lampu, motor, dan aktuator lainnya. Mikrokontroler pada board arduino di program menggunakan bahasa pemrograman arduino (*based on wiring*) dan IDE arduino (*based on processing*). Proyek arduino dapat berjalan sendiri atau juga bisa berkomunikasi dengan *software* yang berjalan pada komputer.

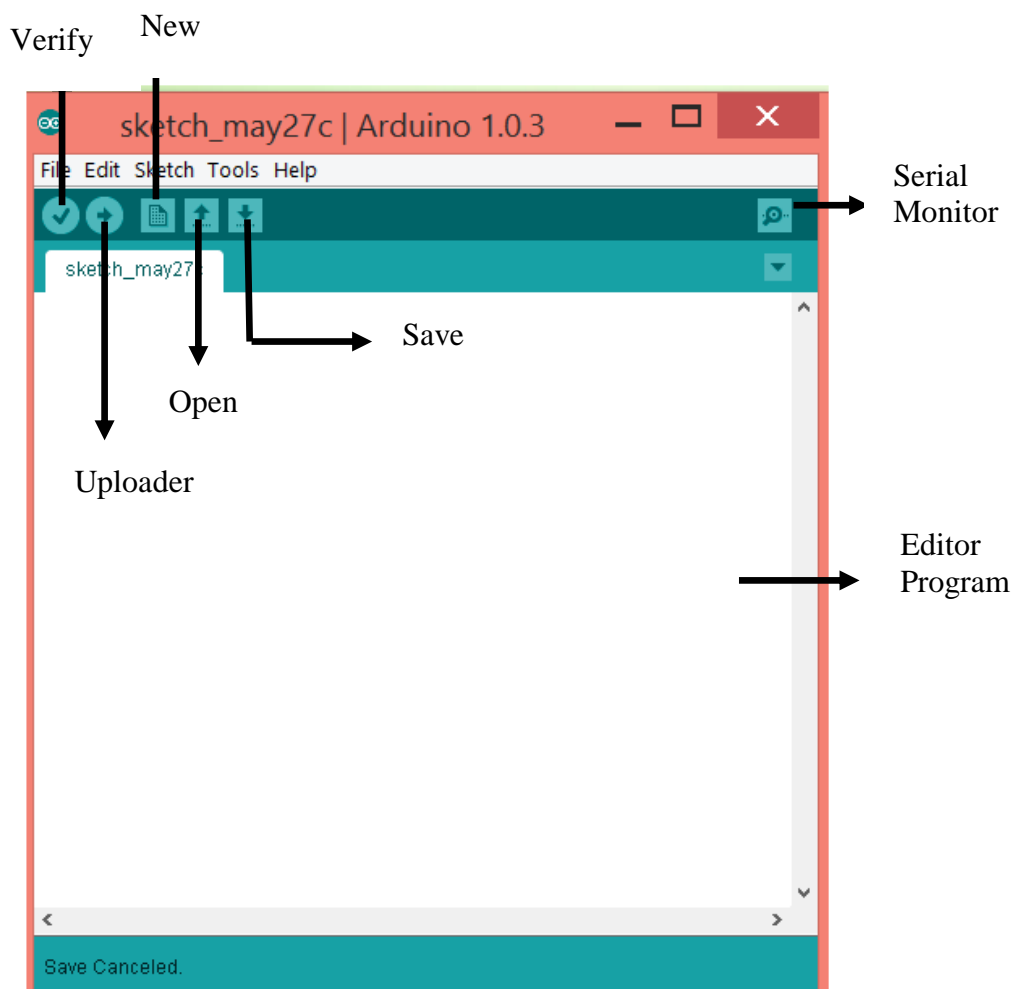
Arduino memakai standar lisensi *open source*, mencakup *hardware* (skema rangkaian, desain PCB), *firmware bootloader*, dokumen, serta perangkat lunak IDE (*Integrated Development Environment*) sebagai aplikasi programmer board Arduino. Setiap modul arduino menggunakan seri mikrokontroler yang berbeda seperti misalnya arduino leonardo yang menggunakan mikrokontroler ATmega328/32U4.

Menggunakan Arduino sangatlah membantu dalam membuat suatu prototyping ataupun untuk melakukan pembatan proyek. Arduino memberikan input output (I/O) yang sudah fix dan bisa digunakan dengan mudah. Arduino dapat digabungkan dengan modul elektro yang lain sehingga proses perakitan jauh lebih efisien.

Para desainer hanya tinggal membuat *software* untuk mendayagunakan rancang hardware yang ada. *Software* jauh lebih mudah untuk dimodifikasi tanpa harus memindahkan kabel. Saat ini arduino sangat mudah dijumpai dan ada beberapa perusahaan yang mengembangkan sistem hardware open source ini. (Artanto, 2012)

### 2.3.2 Software Arduino

*Software* arduino yang digunakan adalah driver dan IDE, walaupun masih ada beberapa *software* lain yang sangat berguna selama pengembangan arduino. IDE atau *Integrated Development Environment* merupakan suatu program khusus untuk suatu komputer agar dapat membuat suatu rancangan atau sketsa program untuk papan Arduino. IDE arduino merupakan *software* yang sangat canggih ditulis dengan menggunakan java. IDE arduino terdiri dari :



**Gambar 2.3 Toolbar Arduino**

(Sumber : penulis, 2019)

Keterangan :

a. Editor Program

Sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa processing.

b. Verify

Mengecek kode sketch yang error sebelum mengupload ke board arduino.

c. Uploader

Sebuah modul yang memuat kode biner dari komputer ke dalam memori di dalam papan arduino.

d. New

Membuat sebuah sketch baru.

e. Open

Membuka daftar sketch pada *sketchbook* arduino.

f. Save

Menyimpan kode sketch pada *sketchbook*.

g. Serial Monitor

Menampilkan data serial yang dikirimkan dari board arduino.

### 2.3.3 Arduino Nano

Arduino Nano adalah salah satu papan pengembangan mikrokontroler yang berukuran kecil, lengkap dan mendukung penggunaan *breadboard*. Arduino Nano diciptakan dengan basis mikrokontroler ATmega328 (untuk Arduino Nano versi 3.x) atau ATmega 168 (untuk Arduino versi 2.x). Arduino Nano kurang lebih memiliki fungsi yang sama dengan Arduino Duemilanove, tetapi dalam



paket yang berbeda. Arduino Nano tidak menyertakan colokan DC berjenis Barrel Jack, dan dihubungkan ke komputer menggunakan port USB Mini-B. Arduino Nano dirancang dan diproduksi oleh perusahaan Gravitech. (



**Gambar 2.4 Arduino Nano**

(Nurliana, 2016)

**a. Sumber Daya Arduino Nano**

Arduino Nano dapat diaktifkan melalui koneksi USB Mini-B, atau melalui catu daya eksternal dengan tegangan belum teregulasi antara 6-20 Volt yang dihubungkan melalui pin 30 atau pin VIN, atau melalui catu daya eksternal dengan tegangan teregulasi 5 volt melalui pin 27 atau pin 5V. Sumber daya akan secara otomatis dipilih dari sumber tegangan yang lebih tinggi. Chip FTDI FT232L pada Arduino Nano akan aktif apabila memperoleh daya melalui USB, ketika Arduino Nano diberikan daya dari luar (Non-USB) maka Chip FTDI tidak aktif dan pin 3.3V pun tidak

tersedia (tidak mengeluarkan tegangan), sedangkan LED TX dan RX pun berkedip apabila pin digital 0 dan 1 berada pada posisi HIGH.

**b. Memory Arduino Nano**

ATmega168 memiliki 16 KB flash memory untuk menyimpan kode (2 KB digunakan untuk *bootloader*); Sedangkan ATmega328 memiliki flash memory sebesar 32 KB, (juga dengan 2 KB digunakan untuk *bootloader*). ATmega168 memiliki 1 KB memory pada SRAM dan 512 byte pada EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM); Sedangkan ATmega328 memiliki 2 KB memory pada SRAM dan 1 KB pada EEPROM.

**c. Input dan Output Arduino Nano**

Masing-masing dari 14 pin digital pada Arduino Nano dapat digunakan sebagai input atau output, dengan menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Semua pin beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor pull-up internal (yang terputus secara default) sebesar 20-50 KOhm. Selain itu beberapa pin memiliki fungsi khusus, yaitu:

- 1) Serial : 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) TTL data serial. Pin ini terhubung ke pin yang sesuai dari chip FTDI USB-to-TTL Serial.

- 2) External Interrupt (Interupsi Eksternal): Pin 2 dan pin 3 ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau perubahan nilai.
- 3) PWM : Pin 3, 5, 6, 9, 10, dan 11. Menyediakan output PWM 8-bit dengan fungsi `analogWrite()`. Jika pada jenis papan berukuran lebih besar (misal: Arduino Uno), pin PWM ini diberi simbol tilde atau “~” sedangkan pada Arduino Nano diberi tanda titik atau strip.
- 4) SPI : Pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI. Sebenarnya komunikasi SPI ini tersedia pada hardware, tapi untuk saat belum didukung dalam bahasa Arduino.
- 5) LED : Pin 13. Tersedia secara built-in pada papan Arduino Nano. LED terhubung ke pin digital 13. Ketika pin diset bernilai HIGH, maka LED menyala, dan ketika pin diset bernilai LOW, maka LED padam.

Arduino Nano memiliki 8 pin sebagai input analog, diberi label A0 sampai dengan A7, yang masing-masing menyediakan resolusi 10 bit (yaitu 1024 nilai yang berbeda). Secara default pin ini dapat diukur/diatur dari mulai Ground sampai dengan 5 Volt, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan fungsi `analogReference()`. Pin Analog 6 dan 7 tidak dapat digunakan sebagai pin digital. Selain itu juga, beberapa pin memiliki fungsi yang dikhususkan, yaitu:

1) I2C : Pin A4 (SDA) dan pin A5 (SCL). Yang mendukung komunikasi I2C (TWI) menggunakan perpustakaan Wire.

Masih ada beberapa pin lainnya pada Arduino Nano, yaitu:

2) AREF : Referensi tegangan untuk input analog. Digunakan dengan fungsi `analogReference()`.

3) RESET : Jalur LOW ini digunakan untuk me-reset (menghidupkan ulang) mikrokontroler. Biasanya digunakan untuk menambahkan tombol reset pada shield yang menghalangi papan utama Arduino.

#### **d. Komunikasi**

Arduino Nano memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, dengan Arduino lain, atau dengan mikrokontroler lainnya. ATmega168 dan ATmega328 menyediakan komunikasi serial UART TTL (5 Volt), yang tersedia pada pin digital 0 (RX) dan pin 1 (TX). Sebuah chip FTDI FT232RL yang terdapat pada papan Arduino Nano digunakan sebagai media komunikasi serial melalui USB dan driver FTDI (tersedia pada *software* Arduino IDE) yang akan menyediakan COM Port Virtual (pada Device komputer) untuk berkomunikasi dengan perangkat lunak pada komputer. Perangkat lunak Arduino termasuk didalamnya serial monitor memungkinkan data tekstual sederhana dikirim ke dan dari papan Arduino. LED RX dan TX yang tersedia pada papan akan berkedip ketika data sedang dikirim atau diterima melalui chip FTDI dan koneksi USB yang terhubung melalui USB komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1).

Sebuah perpustakaan *Software Serial* memungkinkan komunikasi serial pada beberapa pin digital Nano. ATmega168 dan ATmega328 juga mendukung komunikasi I2C (TWI) dan SPI. Perangkat lunak Arduino termasuk perpustakaan *Wire* digunakan untuk menyederhanakan penggunaan bus I2C. Untuk komunikasi SPI, silakan lihat datasheet ATmega168 atau ATmega328.

## 2.4 Motor Stepper

Motor stepper adalah perangkat elektromekanis yang bekerja dengan mengubah pulsa elektronis menjadi gerakan mekanis diskrit. Motor stepper bergerak berdasarkan urutan pulsa yang diberikan kepada motor. Karena itu, untuk menggerakkannya diperlukan pengendali motor stepper yang membangkitkan pulsa-pulsa periodik.



**Gambar 2.5 Motor Stepper**

(Sumber : Penulis, 2019)

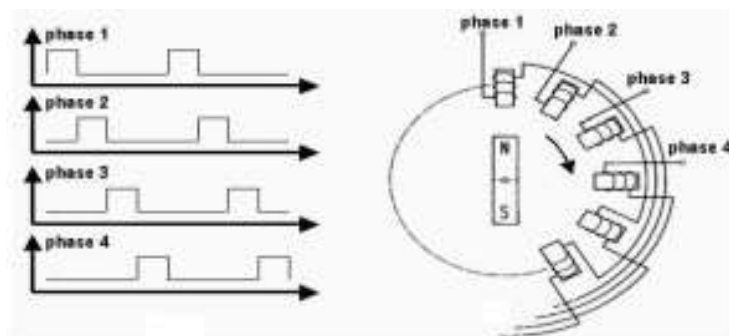
### 2.4.1 Prinsip Kerja Motor Stepper

Motor stepper merupakan perangkat pengendali yang mengkonversikan bit-bit masukan menjadi posisi rotor. Bit-bit tersebut berasal dari terminal-terminal input yang ada pada motor stepper yang menjadi kutub-kutub magnet dalam motor. Bila salah satu terminal diberi sumber tegangan, terminal tersebut akan mengaktifkan kutub di dalam magnet sebagai kutub utara dan kutub yang tidak diberi tegangan sebagai kutub selatan. Dengan terdapatnya dua kutub di dalam motor ini, rotor di dalam motor yang memiliki kutub magnet permanen akan mengarah sesuai dengan kutub-kutub input. Kutub utara rotor akan mengarah ke kutub selatan stator sedangkan kutub selatan rotor akan mengarah ke kutub utara stator.

Prinsip kerja motor stepper mirip dengan motor DC, sama-sama dicatu dengan tegangan DC untuk memperoleh medan magnet. Bila motor DC memiliki magnet tetap pada stator, motor stepper mempunyai magnet tetap pada rotor. Adapun spesifikasi dari motor stepper adalah banyaknya fasa, besarnya nilai derajat per step, besarnya volt tegangan catu untuk setiap lilitan, dan besarnya arus yang dibutuhkan untuk setiap lilitan.

Motor stepper tidak dapat bergerak sendiri secara kontinyu, tetapi bergerak secara diskrit per-step sesuai dengan spesifikasinya. Untuk bergerak dari satu step ke step berikutnya diperlukan waktu dan menghasilkan torsi yang besar pada kecepatan rendah. Salah satu karakteristik motor stepper yang penting yaitu adanya torsi penahan, yang memungkinkan motor stepper menahan

posisinya yang berguna untuk aplikasi motor stepper dalam yang memerlukan keadaan start dan stop.



**Gambar 2.6 Prinsip kerja motor stepper**

(Sumber : Skripsi Guntur Nanda Prakasa, 2017)

## 2.4.2 Karakteristik Motor Stepper

### a. Tegangan

Tiap motor stepper mempunyai tegangan rata-rata yang tertulis pada tiap unitnya atau tercantum pada datasheet masing-masing motor stepper. Tegangan rata-rata ini harus diperhatikan dengan seksama karena bila melebihi dari tegangan rata-rata ini akan menimbulkan panas yang menyebabkan kinerja putarannya tidak maksimal atau bahkan motor stepper akan rusak dengan sendirinya.

### b. Resistansi

Resistansi per lilitan adalah karakteristik yang lain dari motor stepper. Resistansi ini akan menentukan arus yang mengalir, selain itu juga akan mempengaruhi torsi dan kecepatan maksimum dan motor stepper.

### c. Derajat Per Step

Besarnya derajat putaran per step adalah parameter terpenting dalam pemilihan motor stepper karena akan menentukan ukuran langkah gerakan yang paling kecil (resolusi). Tiap-tiap motor stepper mempunyai spesifikasi masing-masing, antara lain  $0.72^\circ$  per step,  $1.8^\circ$  per step,  $3.6^\circ$  per step,  $7.5^\circ$  per step,  $15^\circ$  per step, dan bahkan ada yang  $90^\circ$  per step. Dalam pengoperasiannya kita dapat menggunakan 2 prinsip yaitu full step atau half step. Dengan full step berarti motor stepper berputar sesuai dengan spesifikasi derajat per stepnya, sedangkan half step berarti motor stepper berputar setengah derajat per step dari spesifikasi motor stepper tersebut.

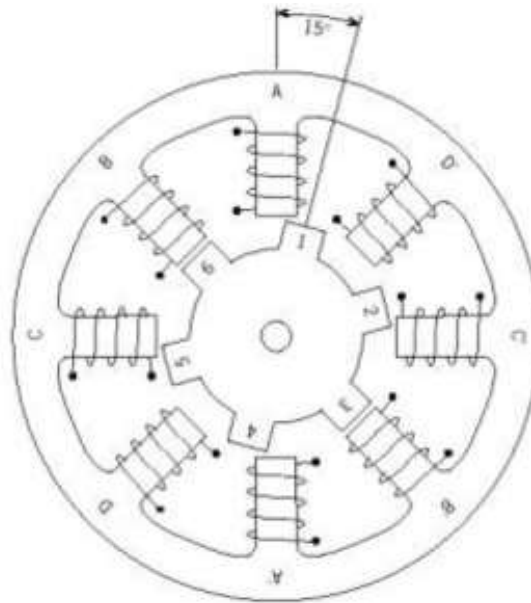
### 2.4.3 Jenis – Jenis Motor Stepper

Berdasarkan struktur rotor dan stator pada motor stepper, maka motor stepper dapat dikategorikan dalam 3 jenis sebagai berikut :

#### a. Motor Stepper Variable Reluctance (VR)

Motor stepper jenis ini telah lama ada dan merupakan jenis motor yang secara struktural paling mudah untuk dipahami. Motor ini terdiri atas sebuah rotor besi lunak dengan beberapa gerigi dan sebuah lilitan stator. Ketika lilitan stator diberi energi dengan arus DC, kutub-kutubnya menjadi termagnetasi. Perputaran terjadi ketika gigi-gigi rotor tertarik oleh kutub-kutub stator. Berikut ini adalah penampang melintang dari motor stepper tipe variable reluctance (VR).



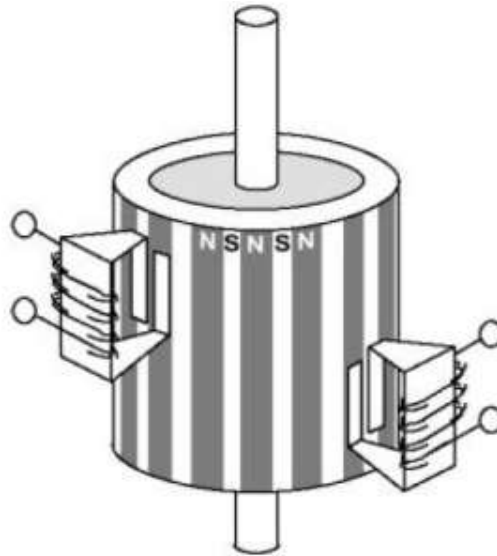


**Gambar 2.7 Motor stepper tipe Variable Reluctance**

(Sumber : Skripsi Guntur Nanda Prakasa, 2017)

**b. Motor Stepper Permanent Magnet (PM)**

Motor stepper jenis ini memiliki rotor yang berbentuk seperti kaleng bundar (tin can) yang terdiri atas lapisan magnet permanen yang diselang-seling dengan kutub yang berlawanan. Dengan adanya magnet permanen, maka intensitas fluks magnet dalam motor ini akan meningkat sehingga dapat menghasilkan torsi yang lebih besar. Motor jenis ini biasanya memiliki resolusi langkah (step) yang rendah yaitu antara 7,50 hingga 150 per langkah atau 48 hingga 24 langkah setiap putarannya. Berikut ini adalah ilustrasi sederhana dari motor stepper tipe permanent magnet.

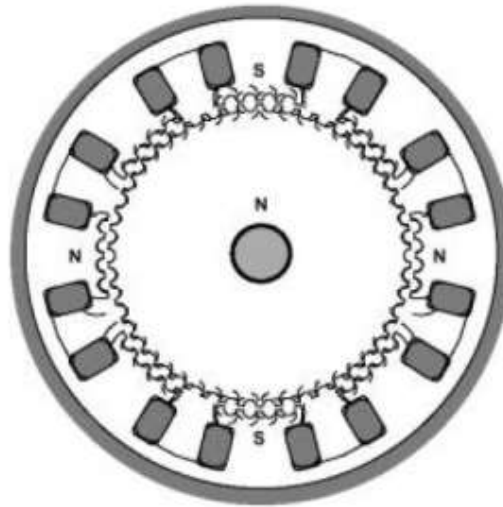


**Gambar 2.8 Motor stepper tipe Permanent Magnet**

(Sumber : Skripsi Guntur Nanda Prakasa, 2017)

**c. Motor Stepper Hybrid (HB)**

Motor stepper tipe hibrid memiliki struktur yang merupakan kombinasi dari kedua tipe motor stepper sebelumnya. Motor stepper tipe hibrid memiliki gigi-gigi seperti pada motor tipe VR dan juga memiliki magnet permanen yang tersusun secara aksial pada batang porosnya seperti motor tipe PM. Motor tipe ini paling banyak digunakan dalam berbagai aplikasi karena kinerja lebih baik. Motor tipe hibrid dapat menghasilkan resolusi langkah yang tinggi yaitu antara 3,60 hingga 0,90 per langkah atau 100-400 langkah setiap putarannya. Berikut ini adalah penampang melintang dari motor stepper tipe hibrid.



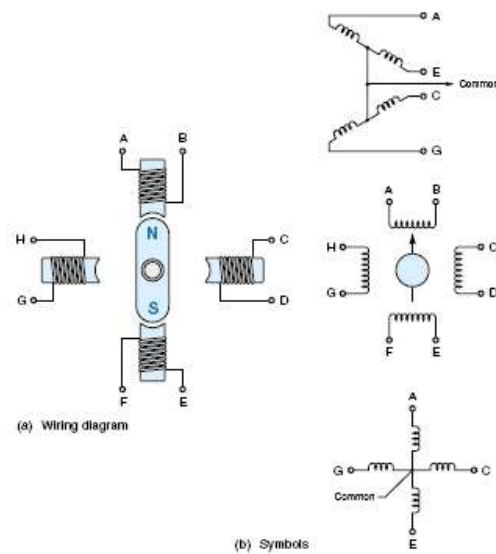
**Gambar 2.9 Motor stepper tipe Hybrid**

(Sumber : Skripsi Guntur Nanda Prakasa, 2017)

- d. Berdasarkan metode perancangan rangkain pengendalinya, motor stepper dapat dibagi menjadi 2 jenis yaitu motor stepper unipolar dan motor stepper bipolar.

#### 1) Motor Stepper Unipolar

Rangkaian pengendali motor stepper unipolar lebih mudah dirancang karena hanya memerlukan satu *switch* / transistor setiap lilitannya. Untuk menjalankan dan menghentikan motor ini cukup dengan menerapkan pulsa digital yang hanya terdiri atas tegangan positif dan nol (*ground*) pada salah satu terminal lilitan (*wound*) motor sementara terminal lainnya dicatu dengan tegangan positif konstan (VM) pada bagian tengah (*center tap*) dari lilitan seperti pada gambar berikut.

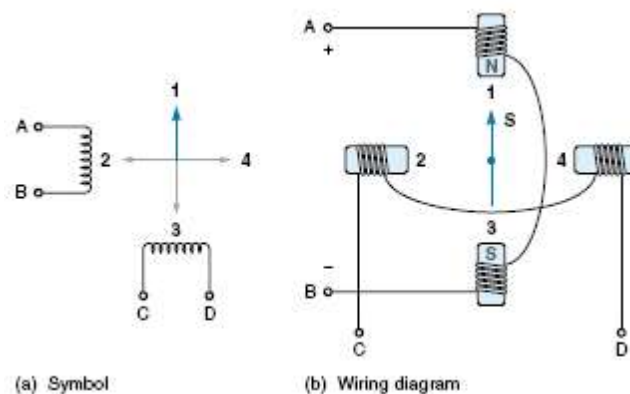


**Gambar 2.10 Motor Stepper dengan lilitan Unipolar**

(Kirilov & Ivanov, 2012)

## 2) Motor Stepper Bipolar

Untuk motor stepper dengan lilitan bipolar, diperlukan sinyal pulsa yang berubah-ubah dari positif ke negatif dan sebaliknya. Jadi pada setiap terminal lilitan (A & B) harus dihubungkan dengan sinyal yang mengayun dari positif ke negatif dan sebaliknya. Karena itu dibutuhkan rangkaian pengendali yang agak lebih kompleks dari pada rangkaian pengendali untuk motor unipolar. Motor stepper bipolar memiliki keunggulan dibandingkan dengan motor stepper unipolar dalam hal torsi yang lebih besar untuk ukuran yang sama.



**Gambar 2.11 Motor Stepper dengan lilitan Bipolar**

(Kirilov & Ivanov, 2012)

## 2.5 UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah “Salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisa & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek” (Rosa dan Shalahuddin 2014).

Dari beberapa penjelasan teori tersebut dapat disimpulkan bahwa UML (*Unified Modeling Language*) adalah bahasa yang sering digunakan untuk membangun sebuah sistem perangkat lunak dengan melakukan penganalisaan desain dan spesifikasi dalam pemrograman berorientasi objek.

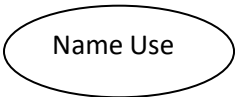
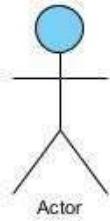

### 1. Use Case Diagram


Use Case Diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informai yang akan dibuat. Use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem

informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

**Tabel 2.1 Simbol – simbol Use Case Diagram**

(Sumber : Rosa dan Shalahuddin, 2013)

Simbol	Deskripsi
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>Use Case</i>
Aktior / <i>Actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informaasi yang akan dibuat itu sendiri, jadi walaupun symbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang: biasanya dinyatakan menggunakan kata benda diawal frase nama aktor
Asosiasi/ <i>Asotiation</i> 	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor
Ekstensi/ <i>extend</i>	Relasi use case tambahan kesebuah use case dinamakan use case yang ditambahkan dapat


<p>-----&lt;&lt;extend&gt;&gt;-----&gt;</p>	<p>berdiri sendiri walau tanpa use case tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya use case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan.</p>
<p>Generalisasi/<i>generalization</i></p> <p style="text-align: center;"></p>	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>


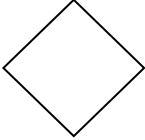


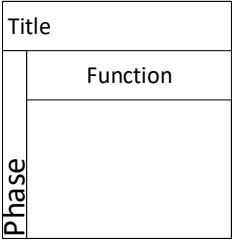
## 2. Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Perlu diperhatikan bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

**Tabel 2.2 Simbol – simbol Activity Diagram**

(Sumber : Rosa dan Shalahuddin, 2013)

Simbol	Deskripsi
<p>Status Awal</p> <p style="text-align: center;"></p>	<p>Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah statusawal</p>

Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penghubungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlan 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

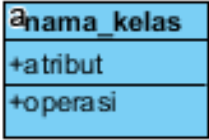






### 3. Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.



**Tabel 2.3 Simbol – simbol Class Diagram**

(Sumber : Rosa dan Shalahuddin, 2013)

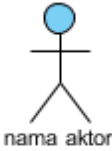
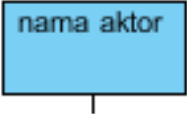

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antar muka/ <i>Interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi/ <i>asotiation</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah / <i>Directed Association</i> 	Relasi antarkelas dengan makna kelas satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan <i>Dependency</i> 	Relasi antar kelas dengan makna kebergantungan antarkelas
Agregasi / <i>Aggregation</i> 	Relas antarkelas dengan makna semuabagian ( <i>whole-part</i> )

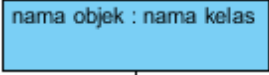





#### 4. Squence Diagram

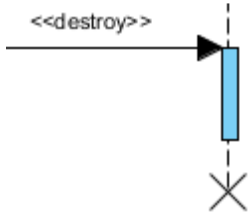
Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Untuk menggambarkan diagram sekuen maka harus diketahui objek - objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

**Tabel 2.4 Simbol – simbol *Sequence Diagram***

(Sumber : Rosa dan Shalahuddin, 2013)

Simbol	Deskripsi
<p>Aktor</p>  <p>Nama Aktor</p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi itu sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor</p>
<p>Garis hidup/<i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>

<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya</p>
<p>Pesan tipe <i>create</i></p>  <p>&lt;&lt;<i>create</i>&gt;&gt;</p>	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
<p>Pesan tipe <i>call</i></p> 	<p>Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi / metode, karena ini memanggil operasi / metode maka operasi / metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
<p>Pesan tipe send</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
<p>Pesan tipe <i>return</i></p> <p>Keluaran</p> 	<p>Menyatakan suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang</p>

	menerima kembalian
<p>Pesan tipe <i>destory</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i></p>

# BAB III

## METODE PENELITIAN

### 3.1 Tahapan Penelitian



**Gambar 3.1 Flow Chart Tahapan Penelitian**

(Sumber : Penulis, 2019)

## **3.2 Metode Pengumpulan Data**

### **1. Penelitian Kepustakaan (*library research*)**

Mencari dan mempelajari berbagai referensi yang mengacu dari berbagai sumber, baik dari buku maupun dari tugas akhir yang dijadikan referensi untuk memperoleh data dan teori-teori yang dibutuhkan untuk mendukung dalam melakukan penelitian meliputi

Studi literatur dilakukan untuk mempelajari teori penunjang yang dibutuhkan dalam prancangan dan pembuatan alat. Teori yang diperlukan yakni

- a. Keseimbangan robot beroda dua
- b. Board Arduino Nano
- c. Sensor Gyroscope dan Accelerometer
- d. Motor DC
- e. Driver motor DC

### **2. Penelitian Laboratorium**

Penelitian laboratorium merupakan penelitian yang dilakukan dalam ruangan tertutup, dimana kelompok eksperimen dijauhkan dari variable pengganggu sebab dapat memengaruhi hasil dari pengujian hubungan sebab akibat.

Penelitian jenis ini dilakukan dalam suatu tempat khusus untuk mengadakan studi-ilmiah dan kerja ilmiah. Tujuan dari penelitian laboratorium untuk ilmu pengetahuan sosial ialah: untuk mengumpulkan data, mengadakan analisa, mengadakan test serta memberikan interpretasi

terhadap sejumlah data, sehingga orang bisa meramalkan kecendrungan gerak dari satu gejala sosial dalam satu masyarakat tertentu. Objek penelitiannya, baik berupa masalah-masalah yang teoritis sifatnya maupun yang praktis, yang diteliti oleh satu tim ahli.

### **3.3 Analisis Sistem Sedang Berjalan**

#### **1. Analisis Sistem yang Berjalan**

Analisis sistem yang berjalan menggambarkan bagaimana sistem yang berjalan pada penelitian yang sudah ada sebelumnya, dimana terdapat perbedaan – perbedaan mulai dari cara pengujian, ketahanan robot dan lama robot dapat menyeimbangkan diri pada bidang datar.

Pada hardware terdapat beberapa perbedaan dimana penelitian sebelumnya menggunakan Mikrokontroler *Arduino Uno*, dan menggunakan suplay daya 10,88 volt.

Dari segi pengujian penelitian sebelumnya lebih terfokus kepada pengujian sudut kemiringan pada robot dan kemiringan sudut lintasan sehingga tidak diketahui berapa lama robot dapat mempertahankan keseimbangannya pada bidang datar dan tidak dapat mengetahui kekuatan rangkaian sistem.

## 2. Analisis Kebutuhan Sistem

### a. Analisis kebutuhan fungsional sistem

Kebutuhan fungsional yang harus dimiliki oleh sistem *self balancing robot* adalah :

- 1) Sistem dapat menggerakkan robot maju, mundur, berbelok ke kiri dan ke kanan.
- 2) Sistem dapat menyeimbangkan posisi berdiri pada saat robot diberi gangguan.
- 3) Sistem dapat menyeimbangkan robot tegak lurus pada bidang datar.

### b. Analisis kebutuhan non – fungsional sistem

Kebutuhan non-fungsional yang harus dimiliki oleh sistem *self balancing robot* adalah :

- 1) Waktu proses koneksi, sehingga dapat mengoptimalkan waktu pengguna sistem.
- 2) Robot dapat bertahan ketika diberi beban dan diberi gangguan.

## 3. Analisis Pemodelan Sistem

Pemodelan sistem bertujuan untuk menampilkan kebutuhan dari sebuah perangkat lunak. Salah satu jenis pemodelan kebutuhan sistem adalah model berbasis skenario, model ini menggambarkan spesifikasi kebutuhan perangkat lunak dari berbagai sudut pandang aktor di dalam perangkat lunak. Pada penelitian ini digunakan *UML* sebagai



bahasa pemodelan untuk mendesain dan merancang *interface self balancing robot*. Model *UML* yang digunakan adalah *use case diagram*, *activity diagram*, dan *sequence diagram*.

#### a. Pemodelan Menggunakan Use Case Diagram

Untuk mengetahui aktor dan *use case* yang akan digunakan, maka dilakukan identifikasi aktor dan identifikasi *use case*. Setelah mendapatkan aktor dan *use case*, maka *use case diagram* dapat digambarkan. Pengidentifikasi aktor dan *use case* pada sistem ini ditunjukkan pada tabel dibawah.

**Tabel 3.1 Deskripsi Aktor**

No.	Aktor	Deskripsi
1	User	Pengguna yang mengaktifkan robot

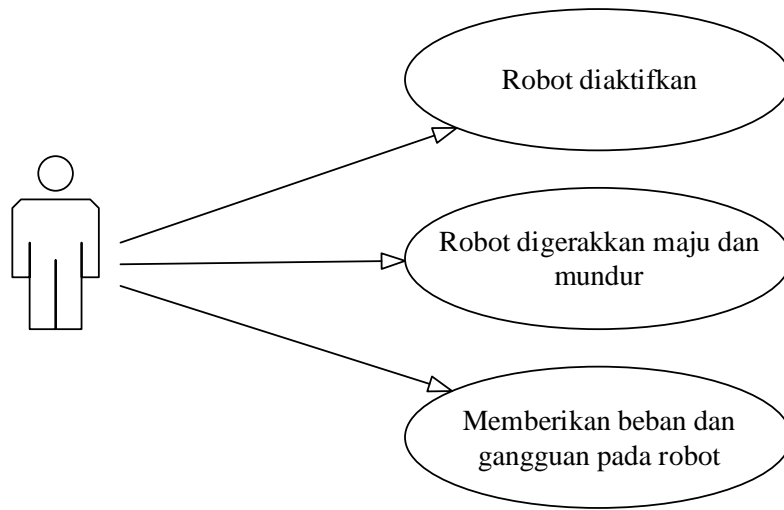
(Sumber : Penulis, 2019)

**Tabel 3.2 ID Use Case**

No.	UC ID	Use Case
1	UC – 1	Mengaktifkan robot
2	UC – 2	Mengarahkan robot untuk bergerak maju dan mundur
3	UC - 3	Memberikan beban dan gangguan pada robot

(Sumber : Penulis, 2019)

*Use case diagram berdasarkan Aktor dan Use case*



**Gambar 3.2 Use Case Diagram Aktor**

(Sumber : Penulis, 2019)

### 1) *Use Case Mengaktifkan Robot*

Pada tabel dibawah akan dijelaskan bagaimana cara kerja sistem untuk mengaktifkan robot. Dapat dilihat pada tabel dibawah yang merupakan dokumentasi naratif dari *use case* gerak robot.

**Tabel 3.3 Use Case Mengaktifkan Robot**

<b>Aktor</b>	<b>Reaksi Sistem</b>
Skenario normal	
1. Menghubungkan konektor <i>battery</i> 9V	2. Mengirim perintah ke mikrokontroler
	3. MPU mencari titik seimbang
Skenario alternatif	

	4. Mematikan koneksi robot
--	----------------------------

(Sumber : Penulis, 2019)

### 2) *Use Case Menggerakkan Robot Maju dan Mundur*

Pada tabel berikut akan dijelaskan bagaimana cara kerja sistem untuk membuat robot bergerak maju dan mundur. Dapat dilihat pada tabel dibawah yang merupakan dokumentasi naratif dari *use case* gerak robot.

**Tabel 3.4 Use Case Menggerakkan Robot maju dan mundur**

<b>Aktor</b>	<b>Reaksi Sistem</b>
Skenario normal	
1. Menghubungkan konektor battery 9V	2. Mengirim perintah ke mikrokontroler
	3. MPU mencari titik seimbang
	4. Menggerakkan robot kedepan dan kebelakang
Skenario alternatif	
	5. Mematikan koneksi robot

(Sumber : Penulis, 2019)

### 3) *Use Case Memberikan beban dan gangguan pada robot*

Pada tabel 3.5 akan dijelaskan bagaimana cara kerja sistem untuk membuat robot tetap seimbang saat diberi beban dan diberi gangguan. Dapat dilihat pada tabel dibawah yang merupakan dokumentasi naratif dari *use case* gerak robot.

**Tabel 3.5 Use Case Memberikan beban dan gangguan pada robot**

<b>Aktor</b>	<b>Reaksi Sistem</b>
Skenario normal	
1. Menghubungkan konektor <i>battery 9V</i>	2. Mengirim perintah ke mikrokontroler
	3.MPU mencari titik seimbang
	4.Memberi beban robot setelah robot seimbang dan memberi gangguan
Skenario alternatif	
	5.Mematikan koneksi robot

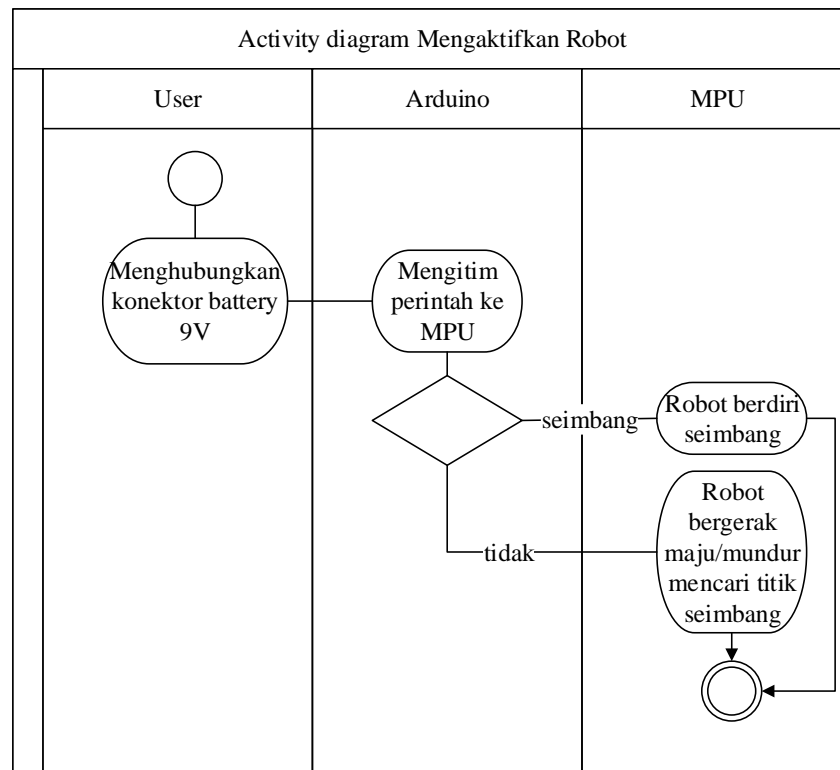
(Sumber : Penulis, 2019)

### **b. Pemodelan Menggunakan *Activity Diagram***

Diagram aktifitas atau *activity diagram* menggambarkan alur kerja (*workflow*) atau aktifitas dari sebuah sistem.

#### **1) *Activity Diagram* Mengaktifkan Robot**

*Activity diagram* mengaktifkan robot maju akan menggambarkan hubungan aliran kerja yang dimulai dari user menghubungkan konektor *battery 9V*, lalu arduino akan mengirim perintah ke MPU untuk menentukan titik seimbang, proses data yang dikirim dan akan dilanjutkan ke stepper motor driver untuk membuat robot bergerak.

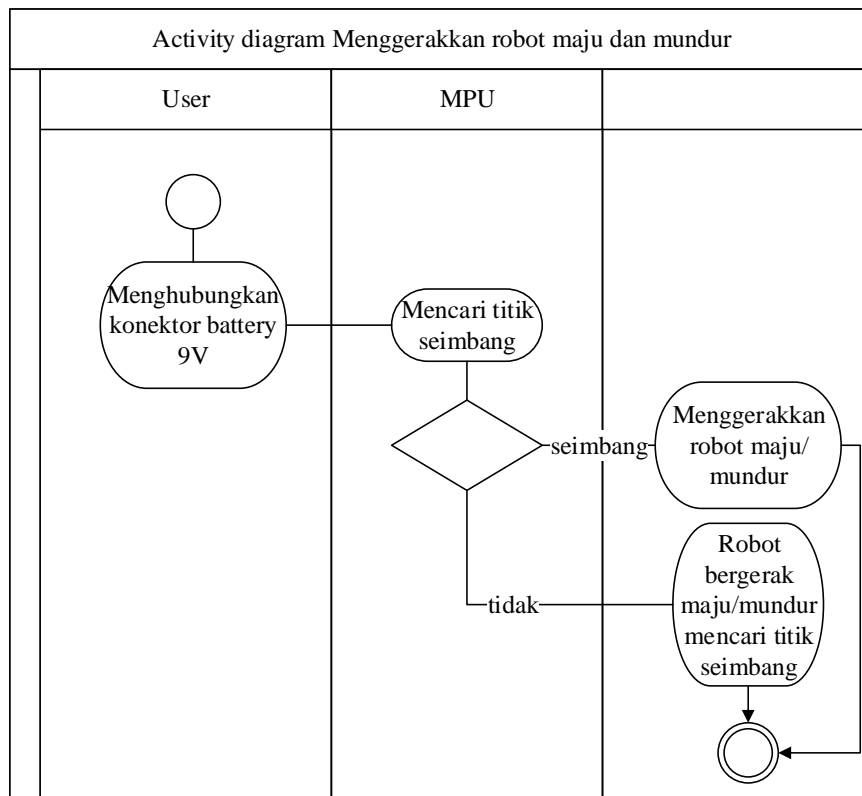


**Gambar 3.3 Activity Diagram Mengaktifkan Robot**

(Sumber : Penulis, 2019)

## 2) Activity Diagram Menggerakkan Robot maju dan mundur

*Activity diagram* menggerakkan robot maju dan mundur akan menggambarkan hubungan aliran kerja yang dimulai dari user menghubungkan konektor *battery* 9V, akan mengirimkan perintah ke *arduino* lalu *arduino* memproses data yang dikirim dan akan dilanjutkan MPU untuk menentukan titik seimbang.

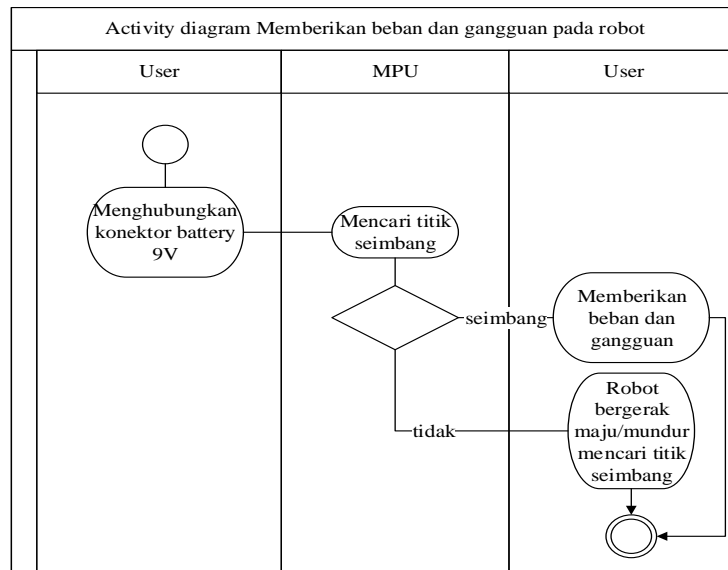


**Gambar 3.4 Activity Diagram Menggerakkan Robot maju dan mundur**

(Sumber : Penulis, 2019)

### 3) Activity Diagram Memberikan beban dan gangguan pada robot

*Activity diagram* memberikan beban dan gangguan pada robot akan menggambarkan hubungan aliran kerja yang dimulai dari user menghubungkan konektor *battery* 9V, akan mengirimkan perintah ke *arduino* lalu *arduino* memproses data yang dikirim dan akan dilanjutkan ke stepper motor driver untuk membuat robot menentukan keseimbangan, setelahnya user akan memberikan beban dan gangguan pada robot sebagai pengujian daya tahan keseimbangan robot.



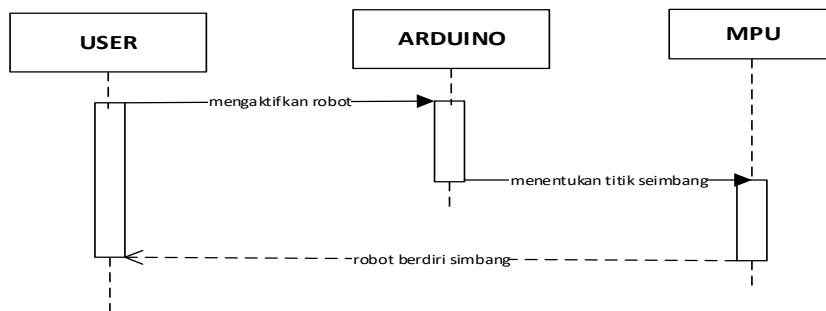
**Gambar 3.5 Activity Diagram Memberikan beban dan gangguan pada robot**

(Sumber : Penulis, 2019)

**c. Pemodelan Menggunakan Sequence Diagram**

**1) Sequence Diagram Mengaktifkan robot**

Pada *sequence* diagram berikut akan digambarkan bagaimana proses sistem mengaktifkan robot.

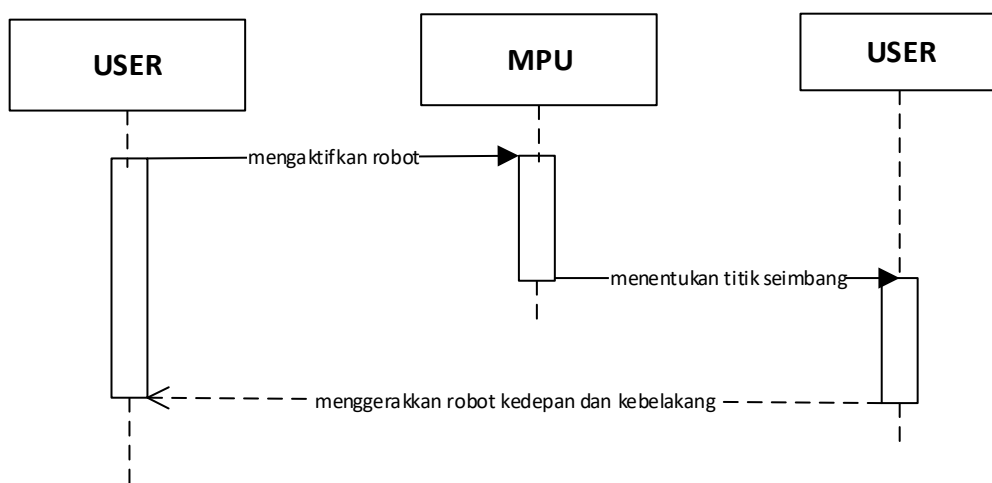


**Gambar 3.6 Sequence Diagram mengaktifkan robot**

(Sumber : Penulis, 2019)

## 2) *Sequence Diagram Menggerakkan Robot maju dan mundur*

Pada *sequence* diagram berikut akan digambarkan bagaimana proses sistem menggerakkan robot maju dan mundur.



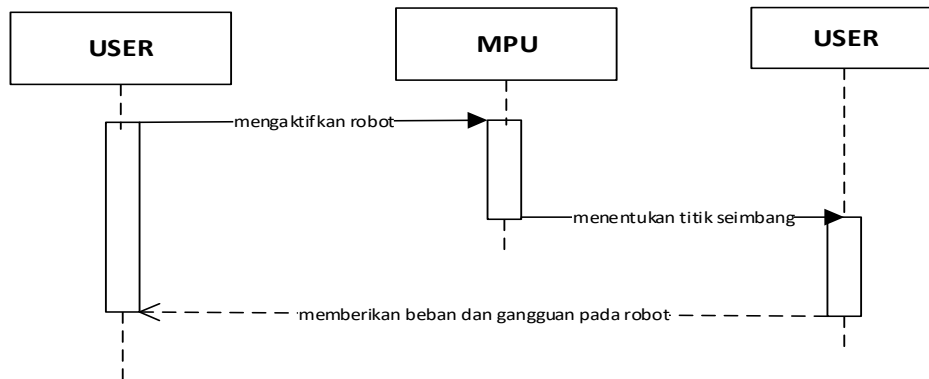
**Gambar 3.7 *Sequence Diagram Menggerakkan robot maju dan mundur***

(Sumber : Penulis, 2019)

## 3) *Sequence Diagram memberikan beban dan gangguan pada robot*

Pada *sequence* diagram berikut akan digambarkan bagaimana robot mempertahankan keseimbangan bila diberi beban dan gangguan, sebagai bahan pengujian untuk daya tahan keseimbangan robot.





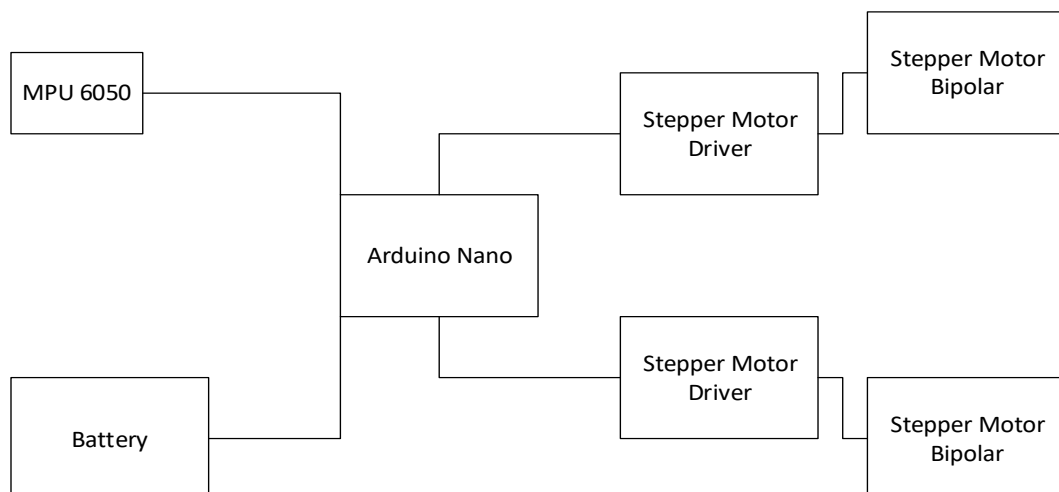
**Gambar 3.8** *Sequence Diagram* memberikan beban dan gangguan pada robot

(Sumber : Penulis, 2019)

### 3.4 Rancangan Penelitian

#### 1. Perangkat Keras

Adapun bentuk rancangan perangkat keras yang dirancang pada penelitian ini seperti gambar dibawah.

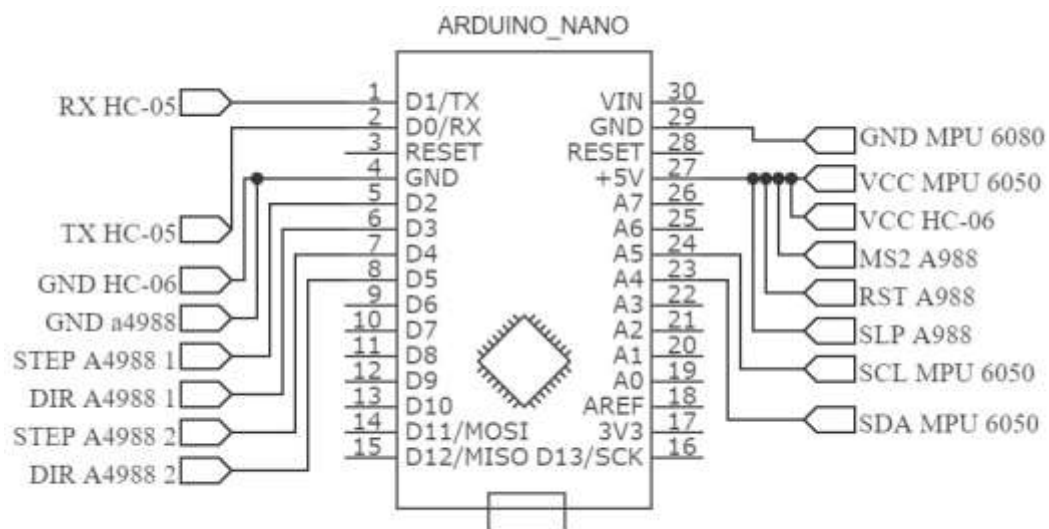


**Gambar 3.9** *Diagram Block* Perancangan *Self Balancing Robot*

(Sumber : Penulis, 2019)

### a. Perancangan Mikrokontroler

Untuk perancangan mikrokontroler penulis menggunakan mikrokontroler arduino nano dan memakai 15 pin. Dimana pin *digital I/O* 1 dan 2 digunakan untuk *modul bluetooth HC - 06* sebagai *RX* dan *TX*, pin *digital* 5 dan 6 digunakan untuk stepper motor driver U3 sebagai *STEP* dan *DIR*, pin *digital* 7 dan 8 digunakan untuk stepper motor driver U4 sebagai *STEP* dan *DIR*. Untuk pin *analog I/O* pin 4 dan 5 digunakan untuk MPU 6050 sebagai *SDA* dan *SCL*, pin *analog* 4 digunakan untuk mengontrol *relay* yang terhubung ke *solenoid valve* dan pin 5 digunakan untuk *input* modul sensor hujan. Pin +5V digunakan untuk daya *modul bluetooth HC – 06* dan *MPU 6050* dan pin *GND* digunakan sebagai *ground* pada modul MPU 6050. Berikut Gambar rancangan mikrokontroler dapat dilihat pada gambar

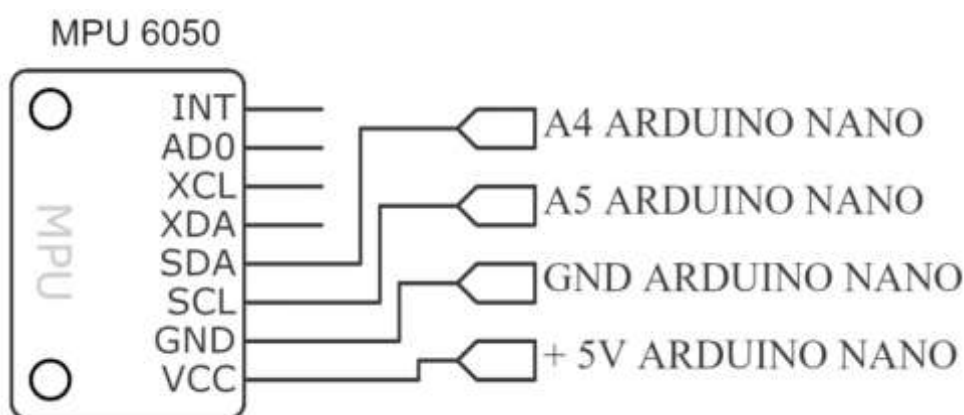


**Gambar 3.10 Rangkaian Mikrokontroler Arduino Nano**

(Sumber : Penulis, 2019)

### b. Rangkaian MPU 6050

Beberapa pin pada modul *MPU 6050* dihubungkan ke pin arduino, yang pertama pin analog 4 dihubungkan pada modul *MPU 6050* yaitu pin SDA, pin analog 5 dihubungkan ke pin SCL pada modul *MPU 6050*. Pin Vcc pada modul *MPU 6050* dihubungkan ke pin +5V arduino dan pin GND pada modul *MPU 6050* dihubungkan ke pin GND arduino. Dibawah merupakan gambar dari modul *MPU 6050*.



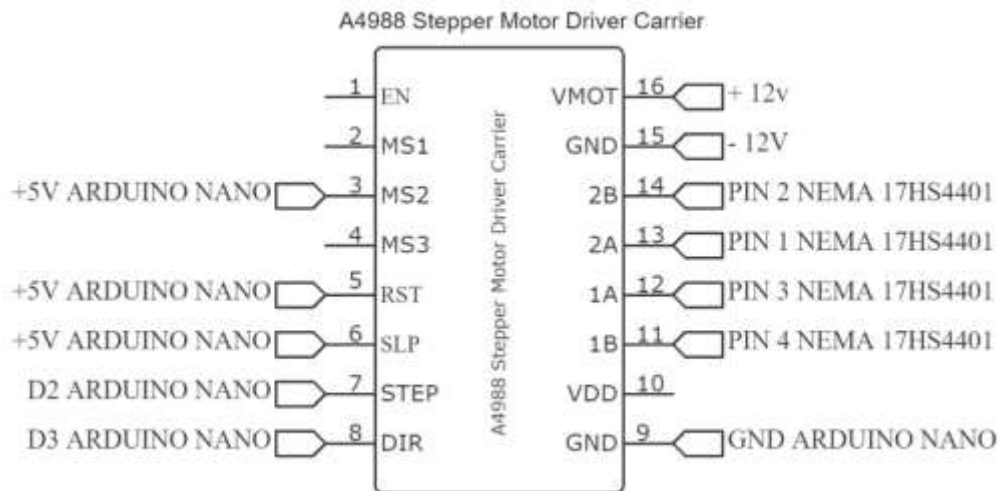
**Gambar 3.11 Rangkaian MPU 6050**

(Sumber : Penulis, 2019)

### c. Rangkaian Stepper Motor Driver

Pada rangkaian *stepper motor driver*, ada beberapa pin yang dihubungkan ke arduino nano pin digital I/O 5 arduino nano dihubungkan pada stepper motor driver U3 ke pin STEP, pin digital 6 dihubungkan pada stepper motor driver U3 ke pin DIR. Pin digital 7 dihubungkan pada stepper motor driver U4 ke pin STEP dan pin digital 8 arduino nano dihubungkan pada pin DIR. Pin SLP, RST dan MS2 pada stepper motor driver dihubungkan ke pin +5V arduino nano dan pin GND pada motor

stepper driver dihubungkan ke pin GND arduino nano. Berikut gambar rangkaian *stepper motor driver*.

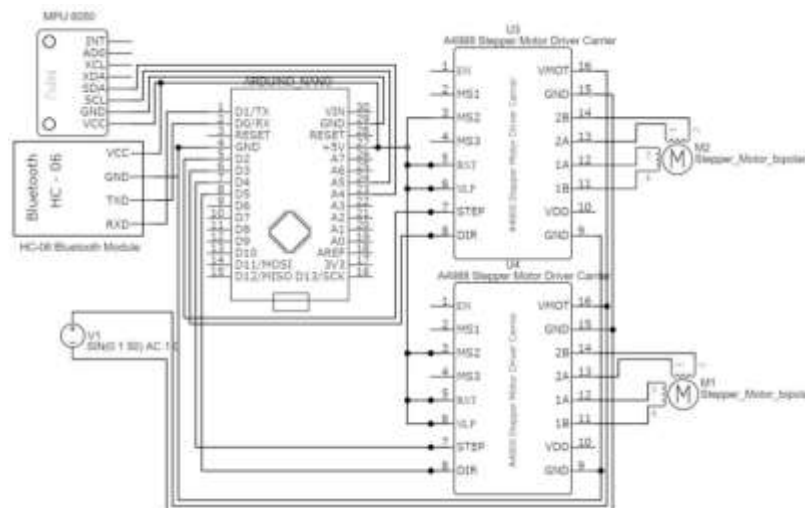


**Gambar 3.12 Rangkaian Stepper Motor Driver Carrier**

(Sumber : Penulis, 2019)

**d. Rangkaian Keseluruhan Sistem**

Berikut gambaran dari perancangan keseluruhan sistem *self balancing robot*.



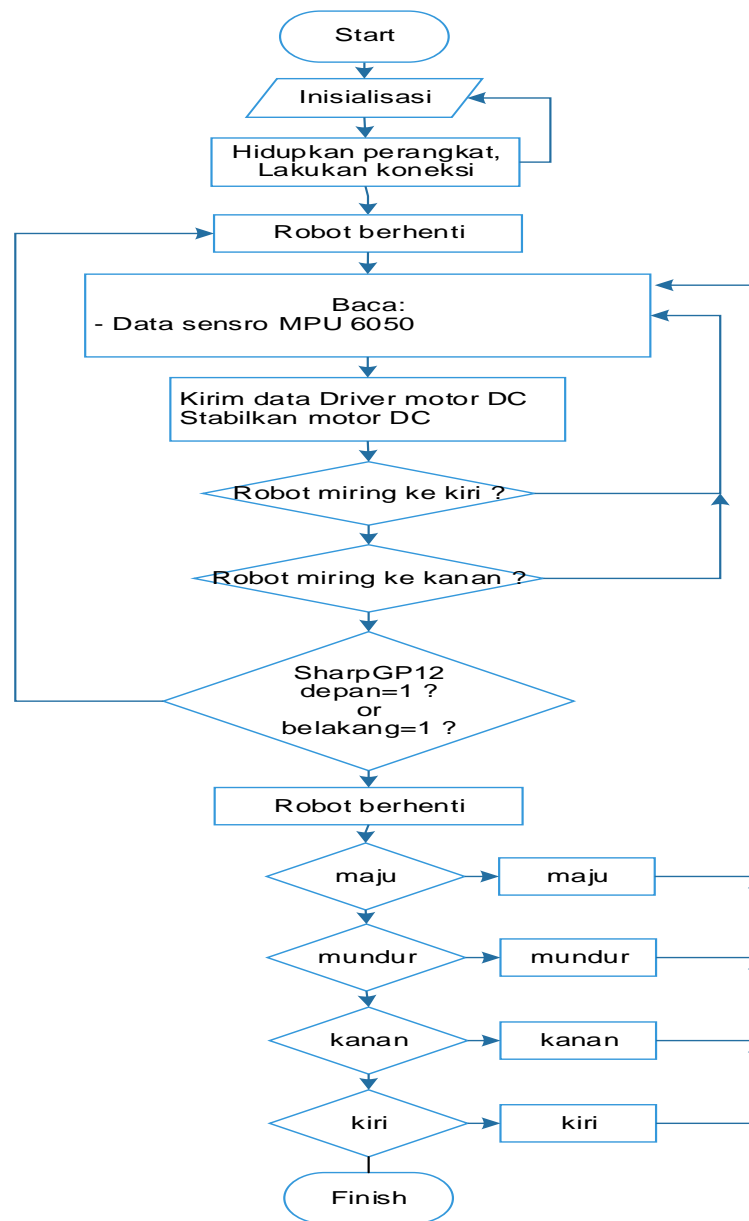
**Gambar 3.13 Rangkaian keseluruhan sistem.**

(Sumber : Penulis, 2019)

## 2. Perangkat Lunak

Perancangan perangkat lunak akan dijelaskan menggunakan diagram alir flowchart agar pembaca lebih memahami bagaimana alur sistem pada *self balancing robot*.

### a. Flowchart Keseluruhan Sistem

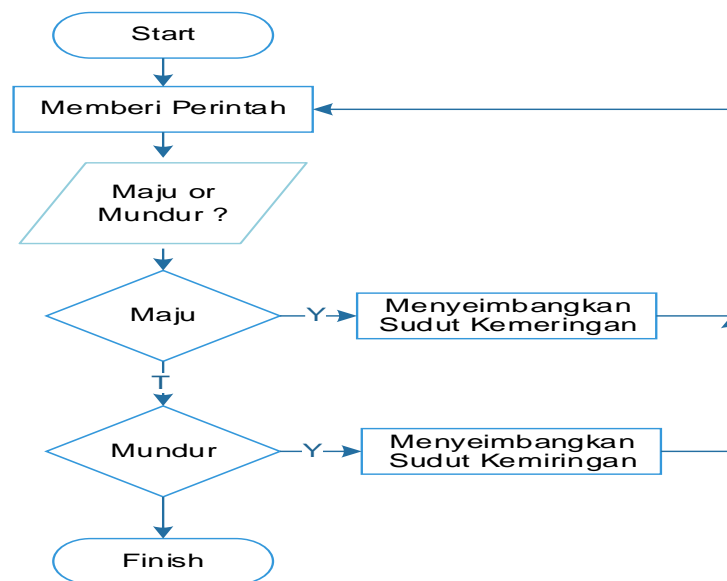


**Gambar 3.14 Flowchart Keseluruhan Sistem.**

Pada gambar diatas dijelaskan alur sistem mulai dari, Start, Hidupkan perangkat dengan menghubungkan tegangan pada rangkaian, Inisialisasi MPU6050 dimaksudkan untuk membaca sudut kemiringan robot, Perangkat menerima data sensor MPU 6050, data sensor SharpGP GP12. Jika sudah seimbang maka robot akan berhenti dan melakukan pengereman untuk mempertahankan keseimbangan robot dan pembacaan sensor akan kembali dilakukan. Jika keadaan robot belum seimbang maka akan dibandingkan sudut kemiringannya, jika robot miring kedepan maka robot bergerak maju dan sebaliknya. Kembali lagi dalam proses pembacaan sensor, jika sensor jarak depan atau sensor jarak belakang menerima data, robot akan berhenti dan menghindari objek,. Finish.

#### b. Flowchart Sensor Accelerometer

Pada flowchart sensor accelerometer akan digambarkan bagaimana alur penyeimbang pada *self balancing robot*.



**Gambar 3.15 Flowchart Sensor Accelerometer**

Gambar 3.11 menunjukkan bagaimana ketika sensor *accelerometer* diberikan perintah untuk maju maka sensor accelerometer akan mendeteksi sudut kemiringan dan akan menyeimbangkan badan dari robot, begitu juga ketika diberi perintah untuk mundur maka hal yang sama akan dilakukan.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1** *Kebutuhan Spesifikasi Minimum Hardware dan Software*

1. *Kebutuhan spesifikasi minimum Hardware*

*Spesifikasi minimum hardware* yang digunakan alat robot bisa berfungsi dengan baik yaitu :

- a. Arduino Nano
- b. Driver Stepper Motor
- c. Sensor Accelerometer MPU 6050
- d. Motor DC
- e. Battery 9 volt

2. *Kebutuhan spesifikasi minimum Software* yang digunakan untuk memprogram robot keseimbangan adalah :

- a. Arduino IDE versi 1.8.2

#### **4.2** *Pengujian Aplikasi dan Pembahasan*

Pada Bab ini *prototype* yang telah di bangun kemudian dilakukan pengujian dan analisis guna untuk mengetahui kinerja sistem *self balancing robot*. Pengujian berupa pengolahan hardware dan software yang telah terintegrasi satu sama lain.



## 1. Pengujian Arduino

*Arduino Nano* merupakan pengendali utama dari hardware yang di buat. Pengujian terhadap *Arduino* ini yaitu untuk mengetahui apakah mikrokontroler ini dapat digunakan dengan baik atau tidak. Cara menguji *hardware* ini yaitu dengan memeriksa setiap *pin input* dan *output* yang terdapat pada *Arduino* yang sebelumnya telah di pasang program pada setiap *pin*. Hasil pengujian menunjukkan bahwa *Arduino* yang sebelumnya telah di program dan disambungkan dengan *hardware* lainnya dapat berjalan.

## 2. Pengujian Accelerometer MPU 6050

Pengujian sensor *accelerometer* bertujuan untuk mengetahui sudut kemiringan pada robot dengan arah putaran motor DC. Sensor *accelerometer* akan terhubung dengan mekanik robot dan arduino nano, data dari sensor *accelerometer* akan dibuat input dan *outputnya* adalah kecepatan dan arah putaran motor DC. Pengujian sensor *accelerometer* dapat dilihat pada gambar dibawah ini.



**Gambar 4.1 Robot berdiri tegak lurus.**

(Sumber : Penulis, 2019)

```

COM4
Balance value: 8337
Printing raw gyro values
Gyro X = 11166 Gyro Y = -280 Gyro Z = -7
Gyro X = 11158 Gyro Y = -271 Gyro Z = -15
Gyro X = 11163 Gyro Y = -286 Gyro Z = -7
Gyro X = 11151 Gyro Y = -290 Gyro Z = -16
Gyro X = 11161 Gyro Y = -265 Gyro Z = -12
Gyro X = 11165 Gyro Y = -290 Gyro Z = -8
Gyro X = 11157 Gyro Y = -291 Gyro Z = 0
Gyro X = 11163 Gyro Y = -269 Gyro Z = -8
Gyro X = 11165 Gyro Y = -280 Gyro Z = -2
Gyro X = 11168 Gyro Y = -297 Gyro Z = -7
Gyro X = 11160 Gyro Y = -283 Gyro Z = -8
Gyro X = 11163 Gyro Y = -266 Gyro Z = -7
Gyro X = 11176 Gyro Y = -271 Gyro Z = 5
Gyro X = 11173 Gyro Y = -291 Gyro Z = -5
  
```

**Nilai titik seimbang robot.**

**Gambar : 4.2 Data keluaran sensor accelerometer robot berdiri tegak lurus.**

(Sumber : Penulis, 2019)

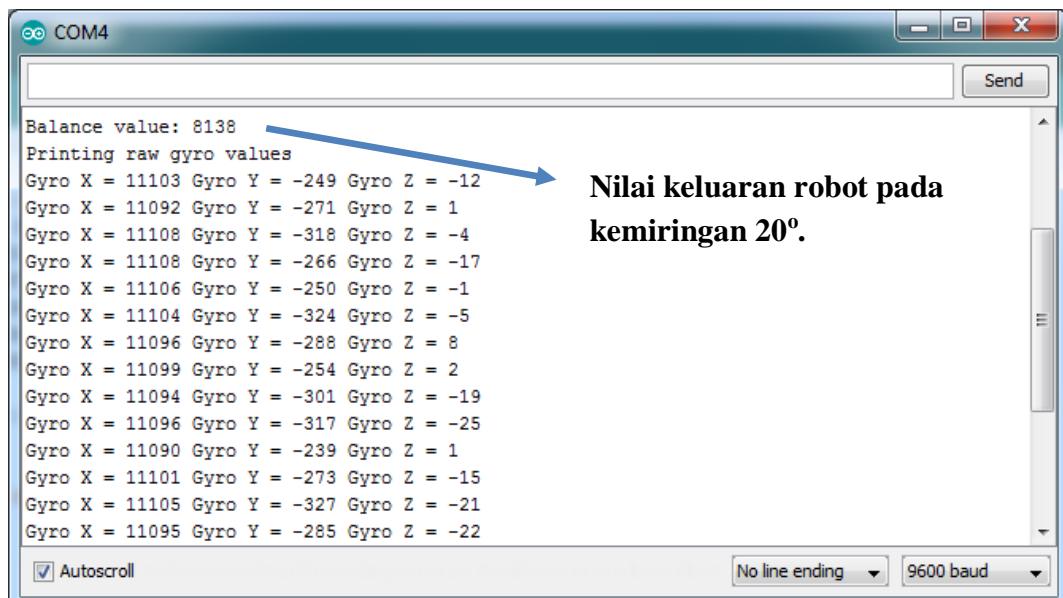
Ketika badan dari robot berdiri tegak seperti yang terlihat pada gambar diatas, maka data keluaran dari sensor akan bernilai titik seimbang seperti yang terlihat pada gambar 4.2. Selanjutnya arah

putaran dari motor DC akan dipengaruhi oleh arah keseimbangan medan jalur. Nilai titik keseimbangan tersebut akan menjadi acuan keseimbangan robot.



**Gambar 4.3 Robot pada kemiringan 20°**

(Sumber : Penulis, 2019)

A screenshot of a COM4 terminal window. The window title is "COM4". It shows a "Send" button in the top right. The main area contains the following text: "Balance value: 8138", "Printing raw gyro values", and a list of 15 lines of gyroscope data. A blue arrow points from the "Balance value: 8138" line to the text "Nilai keluaran robot pada kemiringan 20°." on the right. At the bottom, there are checkboxes for "Autoscroll" (checked) and dropdown menus for "No line ending" and "9600 baud".

```
Balance value: 8138
Printing raw gyro values
Gyro X = 11103 Gyro Y = -249 Gyro Z = -12
Gyro X = 11092 Gyro Y = -271 Gyro Z = 1
Gyro X = 11108 Gyro Y = -318 Gyro Z = -4
Gyro X = 11108 Gyro Y = -266 Gyro Z = -17
Gyro X = 11106 Gyro Y = -250 Gyro Z = -1
Gyro X = 11104 Gyro Y = -324 Gyro Z = -5
Gyro X = 11096 Gyro Y = -288 Gyro Z = 8
Gyro X = 11099 Gyro Y = -254 Gyro Z = 2
Gyro X = 11094 Gyro Y = -301 Gyro Z = -19
Gyro X = 11096 Gyro Y = -317 Gyro Z = -25
Gyro X = 11090 Gyro Y = -239 Gyro Z = 1
Gyro X = 11101 Gyro Y = -273 Gyro Z = -15
Gyro X = 11105 Gyro Y = -327 Gyro Z = -21
Gyro X = 11095 Gyro Y = -285 Gyro Z = -22
```

**Gambar 4.4 Data keluaran sensor accelerometer robot pada kemiringan 20°**

(Sumber : Penulis, 2019)

Pada pengujian sensor accelerometer yang kedua dilakukan dengan kemiringan robot pada titik  $20^\circ$ . Dapat dilihat hasil keluaran nilai pada gambar 4.4, diperoleh nilai 8138 yang berarti nilai keseimbangannya menurun dibandingkan dengan gambar 4.2 yang memiliki nilai tegak lurus 8337.



**Gambar 4.5 Robot pada kemiringan  $40^\circ$**

(Sumber : Penulis, 2019)



```
COM4
Balance value: 7890
Printing raw gyro values
Gyro X = 11088 Gyro Y = -301 Gyro Z = -4
Gyro X = 11083 Gyro Y = -290 Gyro Z = 0
Gyro X = 11075 Gyro Y = -221 Gyro Z = 3
Gyro X = 11085 Gyro Y = -374 Gyro Z = -17
Gyro X = 11093 Gyro Y = -258 Gyro Z = -17
Gyro X = 11103 Gyro Y = -154 Gyro Z = -1
Gyro X = 11100 Gyro Y = -305 Gyro Z = -15
Gyro X = 11109 Gyro Y = -285 Gyro Z = 19
Gyro X = 11105 Gyro Y = -266 Gyro Z = -1
Gyro X = 11098 Gyro Y = -188 Gyro Z = -3
Gyro X = 11111 Gyro Y = -287 Gyro Z = -1
Gyro X = 11103 Gyro Y = -287 Gyro Z = 8
Gyro X = 11097 Gyro Y = -262 Gyro Z = -2
Gyro X = 11098 Gyro Y = -270 Gyro Z = -4
Autoscroll
No line ending
9600 baud
```

**Gambar 4.6 Data keluaran sensor accelerometer robot pada kemiringan  $40^\circ$**

(Sumber : Penulis, 2019)

Dari perbandingan dua pengujian sebelumnya diatas, penulis melakukan pengujian ketiga dengan meletakkan robot dengan posisi kemiringan  $40^\circ$  bertujuan untuk mengetahui nilai keluaran pada sensor accelerometer bila robot dalam keadaan posisi kemiringan  $40^\circ$ . Dapat diketahui nilai keluaran pada gambar 2.6 bernilai 7890.

Dengan begitu didapat sebuah kesimpulan dari pengujian sensor accelerometer, pada setiap bertambah derajat kemiringan pada robot maka akan berkurang nilai keseimbangan pada robot yang didapat saat robot berdiri tegak lurus.

### 3. Pengujian Keseimbangan Robot

Pengujian keseimbangan robot bertujuan untuk mengetahui ketahanan keseimbangan robot saat berdiri, pada penelitian ini dilakukan lima jenis penelitian yang berbeda agar hasil yang didapat dari robot lebih maksimal. Pengujian dilakukan di ruang terbuka/lapangan yang berbidang datar, agar robot memiliki ruang gerak yang lebih luas untuk memperoleh hasil pengujian yang maksimal.

#### a. Pengujian waktu robot menyeimbangkan diri.

**Tabel 4.1 Data pengujian waktu robot menyeimbangkan diri**

No.	Percobaan	Waktu (detik)
1	pertama	9.56
2	Kedua	10.32
3	Ketiga	9.32

4	Keempat	11.23
5	kelima	10.28

(Sumber : Penulis, 2019)

Berdasarkan tabel diatas, pengujian waktu robot menyeimbangkan diri dilakukan sebanyak 5 kali, dengan memposisikan robot dengan posisi berdiri tegak, dengan begitu dapat diketahui rata – rata robot mampu menyeimbangkan diri dengan waktu 10.14 detik.

**b. Pengujian ketahanan keseimbangan robot tanpa gangguan**

**Tabel 4.2 Data pengujian keseimbangan robot.**

No.	Percobaan	Waktu (menit)
1	pertama	2.34
2	Kedua	3.12
3	Ketiga	2.32
4	Keempat	3.13
5	kelima	2.15

(Sumber : Penulis, 2019)

Dari tabel diatas, dilakukan pengujian keseimbangan robot tanpa gangguan apapun sebanyak 5 kali pengujian, dan didapat sebuah hasil keseimbangan lama keseimbangan rata – rata robot dengan waktu 1.99 menit.

**c. Pengujian ketahanan keseimbangan robot ketika diberi gangguan**

**Tabel 4.3 Data pengujian keseimbangan robot diberi gangguan.**

<b>No.</b>	<b>Percobaan</b>	<b>Waktu (menit)</b>
1	pertama	1.26
2	Kedua	1.52
3	Ketiga	1.18
4	Keempat	2.22
5	kelima	2.45

(Sumber : Penulis, 2019)

Diatas merupakan tabel penyajian dari pengujian ketahanan keseimbangan robot ketika diberi gangguan, dimana pengujian ini dilakukan untuk mengetahui tingkat ketahanan robot dan rangkaian robot ketika diberi gangguan ketika sedang menyeimbangkan diri pada bidang datar. Dari 5 kali pengujian diatas diperoleh waktu paling lama robot dapat bertahan ketika diberi gangguan yaitu, 1.26 menit dan waktu paling cepat robot dapat bertahan yaitu, 0.52 menit.

**d. Pengujian keseimbangan robot ketika diberi beban**

**Tabel 4.4 Data pengujian robot keseimbangan diberi beban.**

No.	Percobaan	Jumlah logam	Berat (gram)	Waktu (menit)
1	pertama	4	21	1.06
2	Kedua	8	42	1.00
3	Ketiga	12	63	1.57
4	Keempat	16	73.5	1.58
5	kelima	18	84	1.02

(Sumber : Penulis, 2019)

**e. Pengujian keseimbangan robot ketika diberi beban dan diberi gangguan**

**Tabel 4.5 Data pengujian robot.**

No.	Percobaan	Jumlah logam	Berat (gram)	Waktu (menit)
1	pertama	4	21	0.48
2	Kedua	8	42	1.45
3	Ketiga	12	63	0.37
4	Keempat	16	73.5	1.28
5	kelima	18	84	1.39

(Sumber : Penulis, 2019)

Pengujian keseimbangan robot yang terakhir dilakukan dengan memberi beban dan memberi gangguan pada robot ketika robot diberi beban, dengan begitu dapat diketahui ketahanan rangkaian robot dan juga kekuatan



keseimbangan robot dalam mencari keseimbangan ketika beban yang diberikan tidak sejajar dengan permukaan jalur.

#### 4. Pengujian Robot Secara Keseluruhan

Pengujian sistem secara keseluruhan bertujuan untuk mengetahui kerja dari perangkat keras dan perangkat lunak setelah menghubungkan semua sistem. Mula-mula robot beroda dua diberdirikan posisi tegak lurus terhadap permukaan bidang datar, kemudian di lepaskan untuk melihat kemampuan sistem penyeimbang dalam menjaga keseimbangan robot beroda dua. Parameter yang diamati adalah lama waktu sistem dapat bertahan pada keadaan seimbang. Data pengujian keseimbangan robot dapat dilihat pada tabel 4.6.

**Tabel 4.6 Pengujian Seluruh Sistem**

<b>Pengujian</b>	<b>Waktu (menit)</b>
1	2.34
2	3.12
3	2.32
4	3.13
5	2.15
6	1.12
7	1.52
8	1.18

9	2.22
10	2.45
11	1.06
12	1.00
13	1.57
14	1.58
15	1.02
16	0.48
17	1.45
18	0.37
19	1.28
20	1.39
Rata -rata	1.7 menit

(Sumber : Penulis, 2019)

Dari seluruh pengujian diatas dapat diketahui robot keseimbangan beroda dua yang dirancang dapat mempertahankan keseimbangannya dalam rata - rata waktu 1 sampai 2 menit dengan jenis pengujian yang berbeda – beda.

# **BAB V**

## **PENUTUP**

### **5.1 Simpulan**

Dari perancangan robot dan pengujian, yang telah dilakukan pada robot keseimbangan beroda dua, dapat diambil kesimpulan sebagai berikut :

1. Sistem robot beroda dua yang dirancang dapat menyeimbangkan diri dengan baik, dengan rata – rata waktu keseimbangan 1.7 menit dengan metode pengujian yang berbeda – beda untuk membuktikan ketahanan keseimbangan dan kekuatan rangkaian robot.
2. Sistem *self balancing* robot adalah sistem yang tidak stabil tanpa adanya kontrol sistem sensor *accelerometer*.
3. Mengetahui pengaruh kendali sensor *accelerometer* terhadap keseimbangan robot.

### **5.2 Saran**

Dari perancangan robot keseimbangan ini masih terdapat beberapa kelemahan. Untuk perbaikan sistem robot dan pengembangan lebih lanjut disarankan :

1. Penyempurnaan perancangan pada sistem mekanik, agar kerja robot dalam menyeimbangkan diri lebih maksimal, terutama pada putaran roda agar tidak mempengaruhi saat robot bergerak.

2. Sistem robot keseimbangan hanya dirancang untuk mempertahankan posisi seimbang tegak lurus. Untuk lebih lanjut disarankan untuk membuat control pada *self balancing* robot.
3. Dalam penelitian dan pengembangan selanjutnya, diharapkan untuk bisa diterapkan pada sistem transportasi.

## DAFTAR PUSTAKA

- BOBBY, G., SUSANTO, E., & SURATMAN, F. Y. (2017). Implementasi Robot Keseimbangan Beroda Dua Berbasis Mikrokontroler. *Jurnal Elkomika*, 3(2), 142–160. <https://doi.org/10.26760/elkomika.v3i2.142>
- Hariyanto, E., Lubis, S. A., & Sitorus, Z. (2017). Perancangan Prototipe Helm Pengukur Kualitas Udara. *Konferensi Nasional Teknologi Informasi Dan Komputer (KOMIK)*, 1, 2015–2018.
- Hariyanto, E., Lubis, S. A., & Sitorus, Z. (2017). Perancangan prototipe helm pengukur kualitas udara. *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, 1(1).
- Hartanto, S. (2017). Implementasi fuzzy rule based system untuk klasifikasi buah mangga. *TECHSI-Jurnal Teknik Informatika*, 9(2), 103-122.
- Harumy, T. H. F., & Sulistianingsih, I. (2016). Sistem penunjang keputusan penentuan jabatan manager menggunakan metode mfep pada cv. Sapo durin. In *Seminar Nasional Teknologi Informasi dan Multimedia* (pp. 6-7).
- Havena, M., & Marlina, L. (2018). The Technology of Corn Processing as an Effort to Increase The Income of Kelambir V Village. *Journal of Saintech Transfer*, 1(1), 27-32.
- Indra permana, A. M. I. N. U. D. D. I. N. "Sistem pakar mendeteksi hama dan penyakit tanaman kelapa sawit pada pt. moeis kebun sipare-pare kabupaten batubara." (2013).
- Ii, B. A. B., & Pustaka, T. (2013). *RossumUniversa Robot*. 4–33.
- Kelebihan arduino dari platform hardware mikrokontroler. (2012). (Sumber: Artanto,2012:1) . *Kelebihan Arduino Dari Platform Hardware Mikrokontroler*, 4–27. Retrieved from <https://www.google.com/arduino pdf>
- Kirilov, K., & Ivanov, I. (2012). A programme for determination of codons and codons context frequency of occurrence in sequenced genomes. *Biotechnology and Biotechnological Equipment*, 26(5), 3310–3314. <https://doi.org/10.5504/bbeq.2012.0074>
- Khairul, K., IlhamiArsyah, U., Wijaya, R. F., & Utomo, R. B. (2018, September). Implementasi augmented reality sebagai media promosi penjualan rumaH. In *Seminar Nasional Royal (SENAR)* (Vol. 1, No. 1, pp. 429-434).
- Kurnia, D. (2017). Analisis QoS Pada Pembagian Bandwidth Dengan Metode Layer 7 Protocol, PCQ, HTB Dan Hotspot Di SMK Swasta Al-Washliyah Pasar Senen. *CESS (Journal of Computer Engineering, System and Science)*, 2(2), 102-111.

- Kurniawan, H. (2018). Pengenalan Struktur Baru untuk Web Mining dan Personalisasi Halaman Web. *Jurnal Teknik dan Informatika*, 5(2), 13-19.
- Mariance, U. C. (2018). Analisa dan Perancangan Media Promosi dan Pemasaran Berbasis Web Menggunakan Work System Framework (Studi Kasus di Toko Mandiri Prabot Kota Medan). *Jurnal Ilmiah Core IT: Community Research Information Technology*, 6(1).
- Mayasari, Nova. "Comparison of Support Vector Machine and Decision Tree in Predicting On-Time Graduation (Case Study: Universitas Pembangunan Panca Budi)." *Int. J. Recent Trends Eng. Res* 2.12 (2016): 140-151.
- Muttaqin, Muhammad. "Analisa pemanfaatan sistem informasi e-office pada universitas pembangunan panca budi medan dengan menggunakan metode utaut." *Jurnal Teknik dan Informatika* 5.1 (2018): 40-43.
- Marlina, L., Muslim, M., Siahaan, A. U., & Utama, P. (2016). Data Mining Classification Comparison (Naïve Bayes and C4. 5 Algorithms). *Int. J. Eng. Trends Technol*, 38(7), 380-383.
- Nurliana, S. (2016). *Rancang Bangun Alat Pemberi Isyarat Kecepatan Maksimum Melalui SMS Gateway Berbasis Mikrokontroller Pada Helm*. 4–30.
- Pamungkas, A. P., Elektro, F. T., & Nuswantoro, U. D. (2014). *BALANCING ROBOT BERODA DUA MENGGUNAKAN METODE FUZZY LOGIC Latar Belakang Penelitian dan pengembangan pertama tentang robot dimulai dari tahun 1940-an ketika Argonne National Laboratories di Oak Ridge , Amerika , memperkenalkan sebuah mekanisme robot ic ya*.
- Perwitasari, I. D. (2018). Teknik Marker Based Tracking Augmented Reality untuk Visualisasi Anatomi Organ Tubuh Manusia Berbasis Android. *INTECOMS: Journal of Information Technology and Computer Science*, 1(1), 8-18.
- Puspita, Khairani, and Purwa Hasan Putra. "Penerapan Metode Simple Additive Weighting (SAW) Dalam Menentukan Pendirian Lokasi Gramedia Di Sumatera Utara." *Seminar Nasional Teknologi Informasi Dan Multimedia*, ISSN. 2015.