

## **MODUL IV**

### **SELF-BALANCING MOTORCYCLE**

#### **5.1. Tujuan**

1. Praktikan mampu memprogram *balancing motorcycle* dengan Simulink.
2. Praktikan mampu mengontrol algoritma keseimbangan pada *balancing motorcycle*.
3. Praktikan mampu mengimplementasikan algoritma keseimbangan dengan Simulink pada *balancing motorcycle*.

#### **5.2. Alat dan Bahan**

##### **5.2.1. Komputer**

##### **5.2.2. Matlab**

##### **5.2.3. Arduino Nano 33 IoT**

##### **5.2.4. Arduino Board Nano Motor Carrier**

##### **5.2.5. Self-Balancing Motorcycle**

##### **5.2.6. Li-Ion Baterai**

##### **5.2.7. Micro DC Motor with Encoder**

##### **5.2.8. DC Motor with Encoder**

##### **5.2.9. Servo Motor**

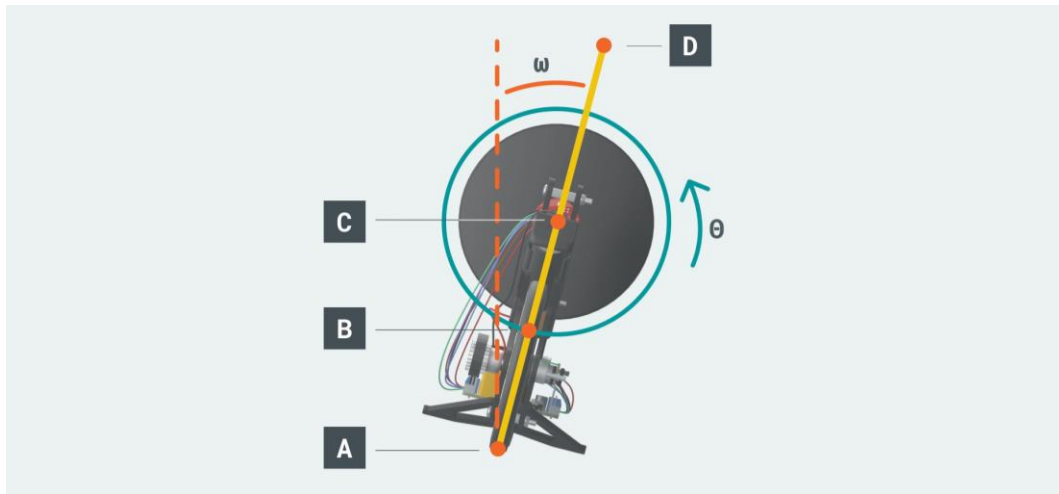
##### **5.2.10. IMU (Inertial Measurement Unit) BN0055**

##### **5.2.11. USB Cable**

#### **5.3. Dasar Teori**

##### **5.3.1. Self-Balancing Robot**

Robot *Self-Balancing* adalah inovasi dari pendulum terbalik yang diletakkan di atas gerobak beroda. Ide dasarnya agar dapat menghasilkan robot beroda dua yang seimbang yaitu dengan mengontrol laju roda searah dengan arah jatuhnya bagian atas robot. Hal tersebut bertujuan agar menjaga pusat gravitasi robot tepat di atas titik pivot.



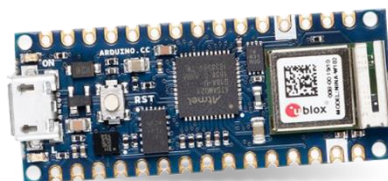
Gambar 5. 1 Teori Self Balancing

Gambar di atas menunjukkan skema teori *self-balancing* pada robot *motorcycle* dengan keterangan sebagai berikut:

- A adalah sumbu rotasi sambungan putar antara batang pendulum dan tanah. Dalam konteks rotasi ini, sumbu A berarti sumbu yang melalui A dan tegak lurus terhadap bidang kertas. Panjang batang bandul adalah panjang rangka sepeda motor.
- B adalah pusat massa batang pendulum. Asumsikan bahwa batang pendulum memiliki kerapatan seragam, yang berarti titik B akan berada di tengah antara A dan D.
- C adalah sumbu rotasi dari sambungan putar antara batang pendulum dan roda inersia. Perhatikan bahwa roda inersia digerakkan oleh motor DC, yang akan dibahas pada praktikum ini.

### 5.3.2. Arduino Nano 33 IoT

Arduino Nano 33 IoT merupakan *development board* buatan dari Arduino yang menggunakan mikrokontroler Arm® Cortex®-M0 32-bit SAMD21 yang terkenal dengan konsumsi daya yang cukup rendah.



Gambar 5. 2 Board Arduino Nano 33 IoT

Arduino Nano 33 IoT juga sudah dilengkapi dengan modul u-blox, the NINA-W10 yang membuat dev board ini memiliki konektivitas WiFi dan Bluetooth yang dapat dioperasikan sampai range 2.4GHz sehingga board ini sangat cocok sekali digunakan untuk keperluan Internet of Things atau IoT.

Selain sudah dilengkapi dengan konektivitas WiFi dan Bluetooth, Arduino Nano 33 IoT ini juga sudah dilengkapi dengan sensor IMU 6 sumbu sehingga bisa digunakan untuk sistem getaran untuk memprediksi gempa ataupun sistem posisi terbang pada pesawat terbang/*drone*.

### 5.3.3. Arduino IDE

Arduino IDE adalah *software* yang digunakan untuk membuat *sketch* pemrograman atau dengan kata lain arduino IDE sebagai media untuk pemrograman pada *board* yang ingin diprogram. Arduino IDE ini berguna untuk mengedit, membuat, meng-*upload* ke *board* yang ditentukan, dan meng-*coding* program tertentu. Arduino IDE dibuat dari bahasa pemrograman JAVA, yang dilengkapi dengan *library* C/C++(*wiring*), yang membuat operasi *input/output* lebih mudah.



Gambar 5. 3 Arduino IDE

### 5.3.4. Matlab

MATLAB (Matrix Laboratory) adalah sebuah bahasa pemrograman tingkat tinggi yang secara khusus digunakan untuk komputasi numerik, pemrograman, dan visualisasi. Perusahaan yang bertanggungjawab atas produk hasil pengembangan dari MATLAB ialah MathWorks. Fungsi utama dari MATLAB ialah untuk melakukan analisis data, mengembangkan algoritma, serta membuat model dan aplikasi. Kinerja MATLAB lebih tinggi dibandingkan bahasa pemrograman

konvensional. Standar variabel elemen pada MATLAB menggunakan konsep larik yang tidak memerlukan proses deklarasi. MATLAB juga dapat mengadakan integrasi dengan bahasa pemrograman dan aplikasi lain seperti C, Java, .NET Framework, dan Microsoft Excel.

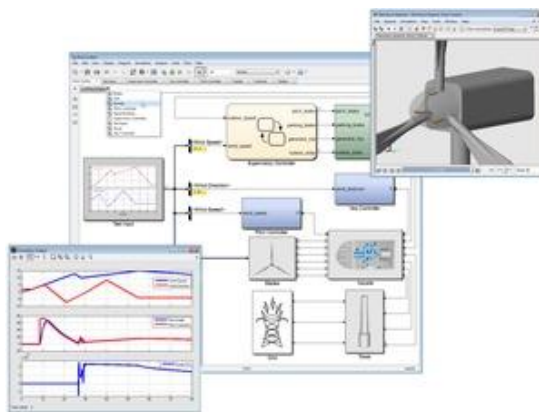


Gambar 5. 4 Logo MATLAB

MATLAB memungkinkan manipulasi matriks, pemplotan fungsi dan data, implementasi algoritma, pembuatan antarmuka pengguna, dan pengantarmuka-an dengan program dalam bahasa lainnya. Meskipun hanya bernuansa numerik, sebuah kotak perkakas (*toolbox*) yang menggunakan mesin simbolik MuPAD, memungkinkan akses terhadap kemampuan aljabar komputer. Sebuah paket tambahan, Simulink menambahkan simulasi grafis multirahang dan Desain Berdasar-Model untuk sistem terlekat dan dinamik.

### 5.3.5. Simulink

Simulink merupakan salah satu komponen dari MATLAB yang berperan sebagai pemrograman grafis. Kegunaan utama dari Simulink adalah untuk membuat simulasi sistem dinamik. Proses simulasi dilakukan menggunakan diagram fungsional yang meliputi blok yang terhubung dengan fungsinya masing-masing secara ekuivalen. Simulink dapat digunakan sebagai sarana pemodelan, simulasi dan analisis dari sistem dinamik dengan menggunakan antarmuka pengguna grafis.



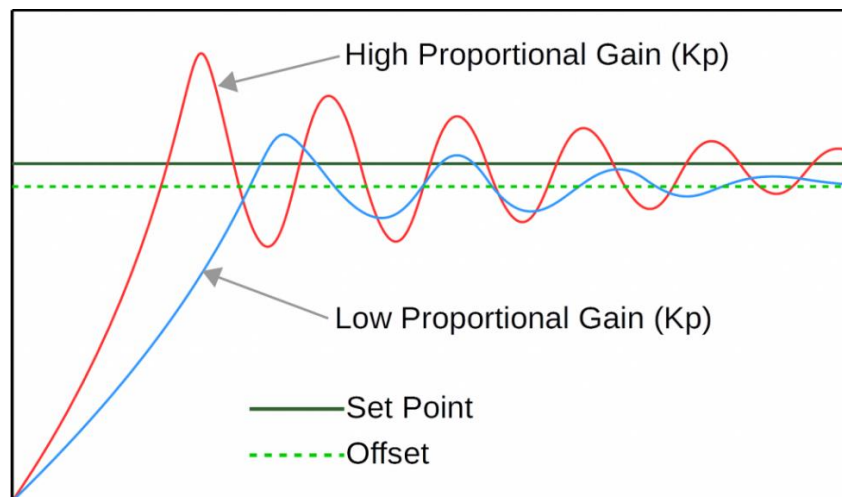
Gambar 5. 5 Simulink

Simulink terdiri dari beberapa kumpulan kotak perkakas yang dapat digunakan untuk analisis sistem linier dan non-linier. Beberapa pustaka yang sering digunakan dalam sistem kontrol antara lain *math*, *sinks*, dan *sources*.

### 5.3.6. PID Controller

Pengontrol PID (Pengontrol Proporsional-Integral-Derivatif) adalah mekanisme umpan balik loop kontrol yang banyak digunakan dalam sistem kontrol industri. Kontroler ini secara terus menerus menghitung nilai kesalahan sebagai perbedaan antara setpoint yang diinginkan (target) dan variabel proses yang diukur. Pengontrol menerapkan sinyal kontrol untuk mendekatkan variabel proses ke setpoint dengan mempertimbangkan tiga komponen:

#### 1. Proportional



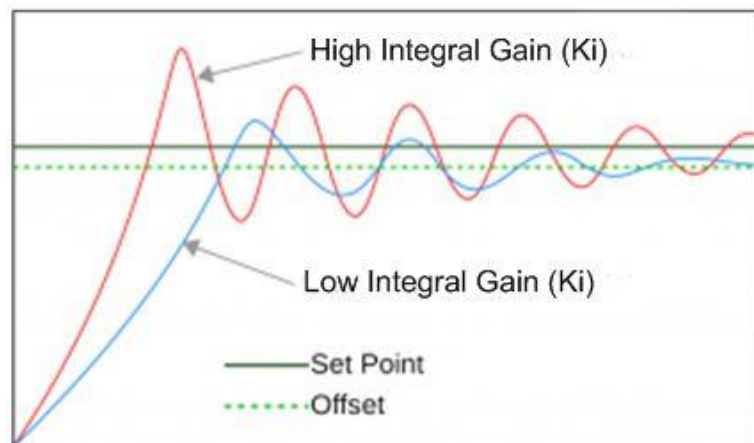
Proporsional menghasilkan nilai output yang sebanding dengan nilai kesalahan saat ini. Ini menentukan respons pengontrol terhadap kesalahan saat ini. Penguatan proporsional yang lebih tinggi menghasilkan perubahan yang lebih besar pada output untuk perubahan kesalahan yang diberikan.

Ciri-ciri pengontrol proporsional :

1. Jika nilai  $K_p$  kecil, pengontrol proporsional hanya mampu melakukan koreksi kesalahan yang kecil, sehingga akan menghasilkan respon sistem yang lambat (menambah rise time).
2. Jika nilai  $K_p$  dinaikkan, respon/tanggapan sistem akan semakin cepat mencapai keadaan mantapnya (mengurangi rise time).

3. Namun jika nilai  $K_p$  diperbesar sehingga mencapai harga yang berlebihan, akan mengakibatkan sistem bekerja tidak stabil atau respon sistem akan berosilasi.
4. Nilai  $K_p$  dapat diset sedemikian sehingga mengurangi steady state error, tetapi tidak menghilangkannya.

## 2. Integral

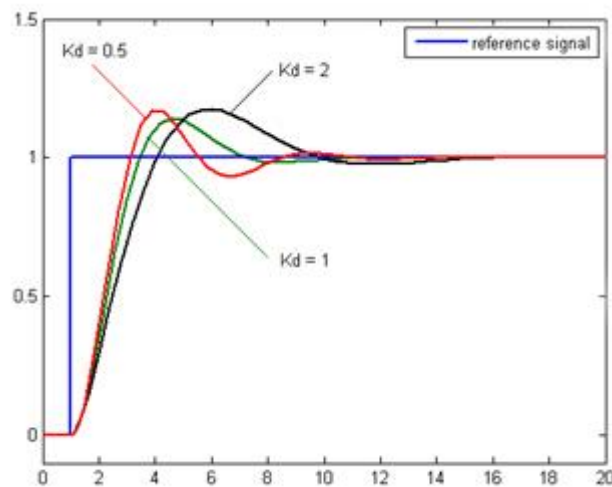


Istilah integral berhubungan dengan akumulasi kesalahan masa lalu dari waktu ke waktu. Ini membantu menghilangkan kesalahan kondisi tunak dengan mengintegrasikan kesalahan dari waktu ke waktu. Ini sangat berguna untuk proses yang perlu mencapai dan mempertahankan setpoint tertentu.

Ciri-ciri pengontrol integral :

1. Keluaran pengontrol integral membutuhkan selang waktu tertentu, sehingga pengontrol integral cenderung memperlambat respon.
2. Ketika sinyal kesalahan berharga nol, keluaran pengontrol akan bertahan pada nilai sebelumnya.
3. Jika sinyal kesalahan tidak berharga nol, keluaran akan menunjukkan kenaikan atau penurunan yang dipengaruhi oleh besarnya sinyal kesalahan dan nilai  $K_i$ .
4. Konstanta integral  $K_i$  yang berharga besar akan mempercepat hilangnya offset. Tetapi semakin besar nilai konstanta  $K_i$  akan mengakibatkan peningkatan osilasi dari sinyal keluaran pengontrol.

### 3. Differential



Istilah turunan memprediksi kesalahan di masa depan berdasarkan tingkat perubahannya. Ini membantu mengurangi overshoot dan meredam osilasi dengan mengantisipasi perilaku kesalahan di masa depan. Term ini bereaksi terhadap laju perubahan error dan dapat dianggap sebagai elemen "prediktif".

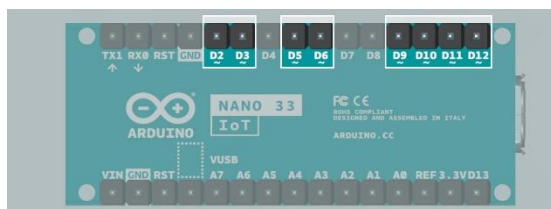
Ciri-ciri pengontrol derivatif :

1. Pengontrol tidak dapat menghasilkan keluaran jika tidak ada perubahan pada masukannya (berupa perubahan sinyal kesalahan)
2. Jika sinyal kesalahan berubah terhadap waktu, maka keluaran yang dihasilkan pengontrol tergantung pada nilai  $K_d$  dan laju perubahan sinyal kesalahan.
3. Pengontrol diferensial mempunyai suatu karakter untuk mendahului, sehingga pengontrol ini dapat menghasilkan koreksi yang signifikan sebelum pembangkit kesalahan menjadi sangat besar. Jadi pengontrol diferensial dapat mengantisipasi pembangkit kesalahan, memberikan aksi yang bersifat korektif dan cenderung meningkatkan stabilitas sistem.
4. Dengan meningkatkan nilai  $K_d$ , dapat meningkatkan stabilitas sistem dan mengurangi *overshoot*.

### 5.3.7. PWM (Pulse Width Modulation) Microcontrollers

Kebanyakan mikrokontroler memiliki kemampuan untuk menghasilkan sinyal PWM (*Pulse Width Modulation*) khusus. Biasanya, tidak semua pin digital memiliki opsi sekunder ini, jadi Anda harus memeriksa lembar data jika Anda memerlukan sinyal PWM untuk melihat pin mana yang memiliki kemampuan ini. Ada juga perpustakaan yang dapat Anda gunakan untuk menghasilkan sinyal PWM "perangkat lunak" di pin GPIO (*General Purpose Input/Output*).

Pada papan Arduino, pin PWM dilambangkan dengan simbol “tilde” ~ di sebelah nomor pin. Pada Arduino Nano 33 IoT, pinnya adalah 2, 3, 5, 6, 9, 10, 11 dan 12. Dalam modul ini sinyal PWM digunakan untuk mengontrol kecepatan semua motor DC.



Gambar 5. 6 Pin PWM pada Arduino Nano 33 IoT

### 5.3.8. Motor DC Encoder

Motor DC (*Direct Current*) adalah jenis motor yang akan menyebabkan poros motor berputar di sekitar sumbu longitudinalnya ketika menerapkan arus listrik di antara pin terminalnya. Dengan demikian, motor DC adalah jenis aktuator yang mengubah arus listrik menjadi gerak rotasi.



Gambar 5. 7 Motor DC

Ada dua bagian di dalam motor: rotor (poros adalah bagian dari ini) dan stator. Ketika melihat penampang motor maka dapat dilihat bahwa rotor adalah bagian yang bergerak dan stator adalah bagian yang statis. Stator dan rotor menggunakan magnet permanen dan elektromagnet. Tergantung pada jenis

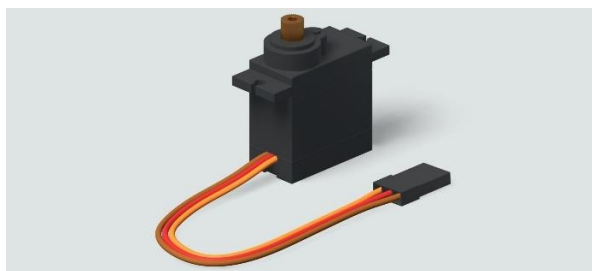


motornya, stator bisa menjadi magnet permanen sedangkan rotor adalah elektromagnet, atau sebaliknya. Menyalakan elektromagnet menciptakan gaya tarik-menarik dan gaya tolak menolak yang membuat motor berputar.

Motor DC bekerja dengan cara berputar ketika diterapkan tegangan DC melalui dua pin terminalnya. Variasi kecepatan motor dapat diatur dengan mengubah level tegangan. Selain itu, motor dapat berjalan bebas di kedua arah hanya dengan membalikkan arah arus. Motor DC saja tidak terlalu bagus untuk pergerakan yang presisi, tetapi dapat memberikan kecepatan rotasi yang sangat tinggi.

#### **5.3.9. Motor Servo Stepper**

Motor servo adalah aktuator yang memungkinkan kontrol posisi (sudut) atau kecepatan sudut yang tepat dari mikrokontroler. Mereka memiliki sirkuit kontrol tertanam di dalam motornya. Sirkuit kontrol ini bisa analog atau digital, dan ini ditentukan oleh jenis fungsi yang dirancang untuk dilakukan oleh motor. Dari perspektif fungsional, ada dua jenis motor servo: servo standar dan servo rotasi kontinu. Dalam modul ini hanya akan menggunakan motor Servo standar untuk berputar dalam kisaran 0 - 180.



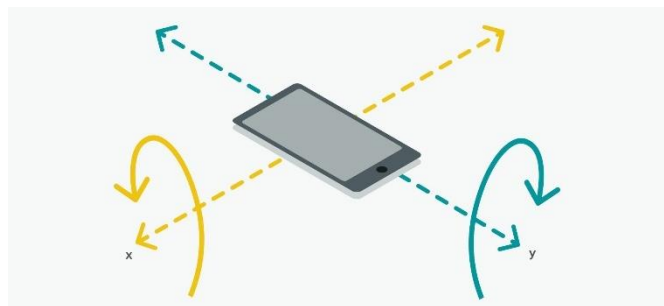
Gambar 5. 8 Motor Servo

Servo standar tidak dapat terus berputar seperti motor DC, sehingga tidak dapat digunakan untuk menggerakkan roda. Namun, ini dapat digunakan untuk memindahkan sesuatu ke depan dan ke belakang pada sudut tertentu dengan akurasi tinggi. Misalnya, jika digunakan pada tutup kotak, dapat membuka tutupnya dengan memutar ke posisi 90°. Kemudian dapat menutup tutupnya dengan kembali ke posisi 0°.

### 5.3.10. IMU (Inertial Measurement Unit) BN0055

IMU (*Inertial Measurement Unit*) adalah perangkat yang digunakan untuk mengukur orientasi dan akselerasi perangkat menggunakan kombinasi akselerometer, giroskop, dan magnetometer. Mereka biasanya digunakan di ponsel, perangkat yang dapat dikenakan, robot otonom, dan pesawat terbang.

IMU adalah modul elektronik yang mengukur perubahan percepatan linier, rotasi sudut, dan, dalam beberapa kasus, medan magnet di sekitar modul. Dengan demikian, *chip* IMU mencakup berbagai sensor, akselerometer, giroskop, dan magnetometer, dalam satu wadah.



Gambar 5. 9 Ilustrasi penggunaan IMU

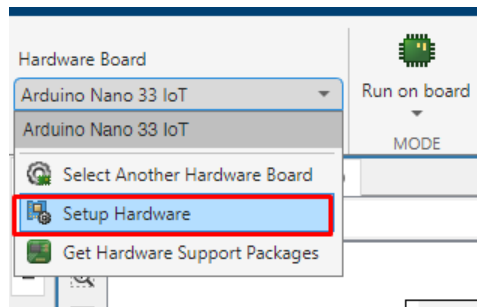
Data mentah yang berasal dari sensor di IMU kemudian diproses dan digabungkan menjadi informasi lain yang lebih mudah digunakan dalam proyek kami. Informasi yang kami ekstrak disebut sebagai: *pitch*, *roll*, dan *yaw*. Masing-masing pengukuran ini berkorelasi dengan salah satu sumbu X, Y, atau Z.

Sensor BNO055 adalah solusi *System in Package* (SiP) yang mengintegrasikan akselerometer triaksial 14-bit, giroskop 16-bit triaksial *close-loop* akurat, sensor geomagnetik triaksial, dan mikrokontroler 32-bit yang menjalankan perangkat lunak FusionLib BSX3.0. Sensor ini secara signifikan lebih kecil daripada solusi yang sebanding. Dengan mengintegrasikan sensor dan fusi sensor dalam satu perangkat, BNO055 memudahkan integrasi, menghindari solusi *multivendor* yang kompleks dan dengan demikian menyederhanakan inovasi, misalnya aplikasi baru seperti perangkat keras IoT.

## 5.4. Langkah Kerja

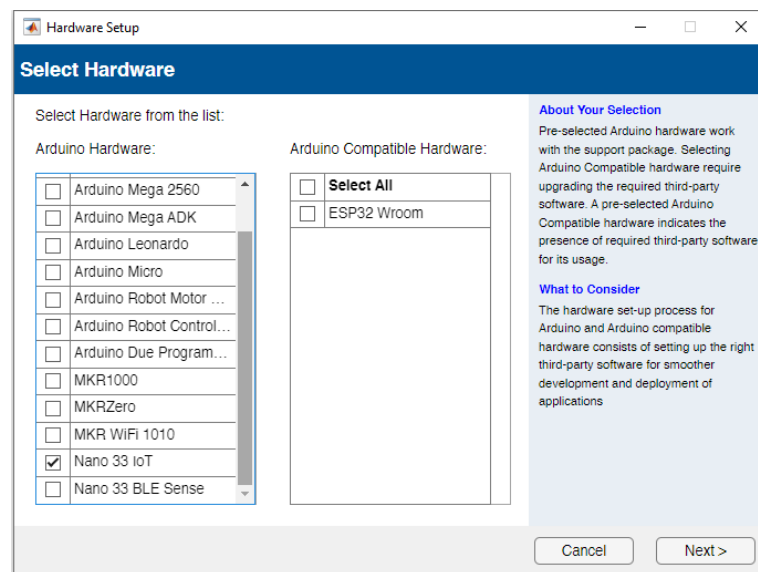
### 5.4.1.1. Konfigurasi Simulink

1. Untuk memastikan Arduino Nano terbuang cek pada setup hardware dapat dilakukan di menu ini



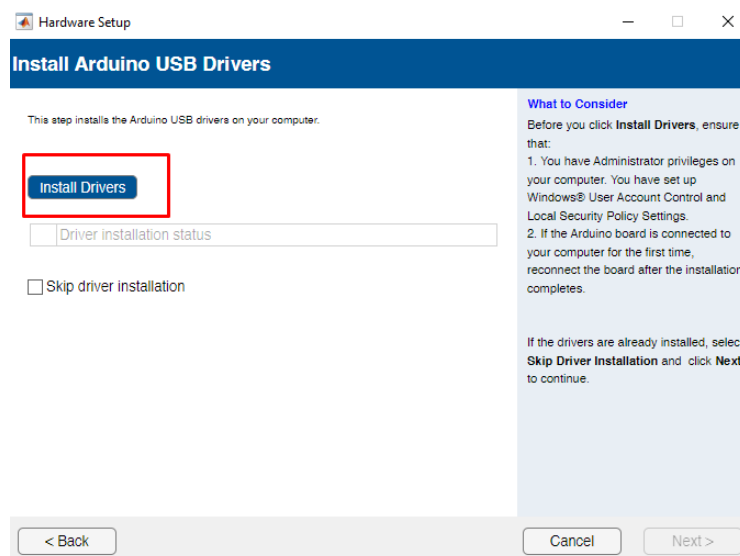
Gambar 5. 10 Setting setup hardware

2. Pilih Nano 33 IoT selanjutnya Next dan Next lagi



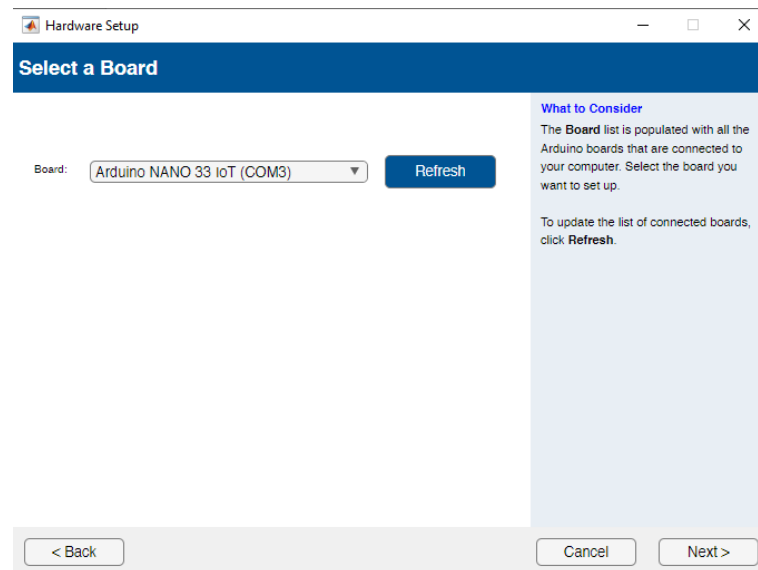
Gambar 5. 11 Hardware Setup

3. Jika belum ter install klik install



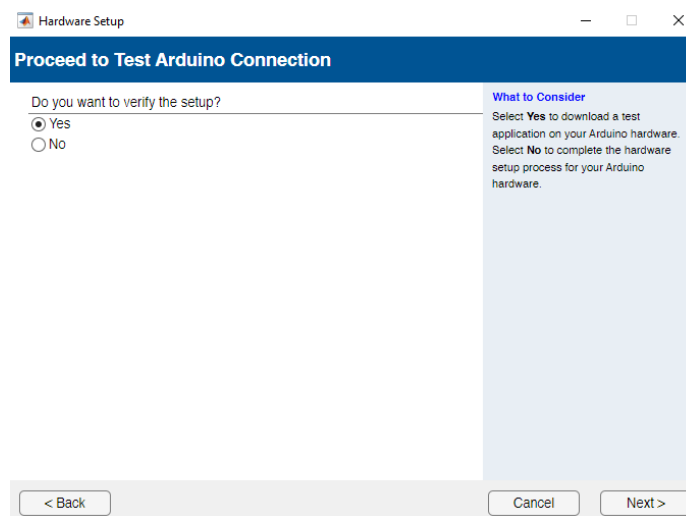
Gambar 5. 12 Install Drivers

4. Setelah itu bakal terdeteksi secara otomatis *board* yang akan dipakai



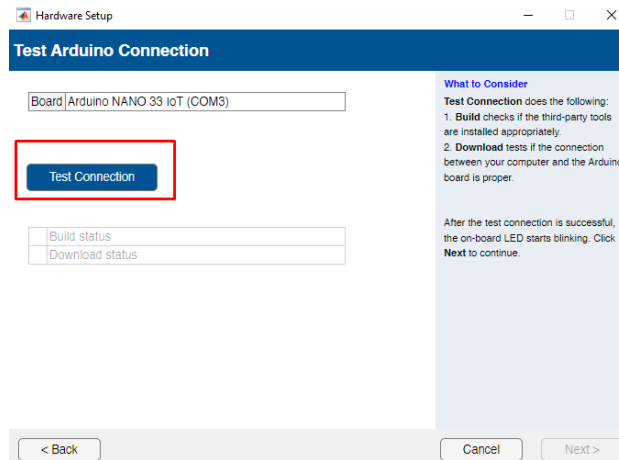
Gambar 5. 13 Pemilihan Board

5. Kemudian verif setup



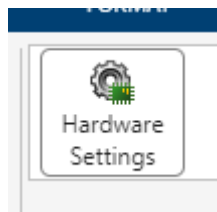
Gambar 5. 14 Test Arduino Connection

6. Klik Test Connection setelah OK semua kemudian next



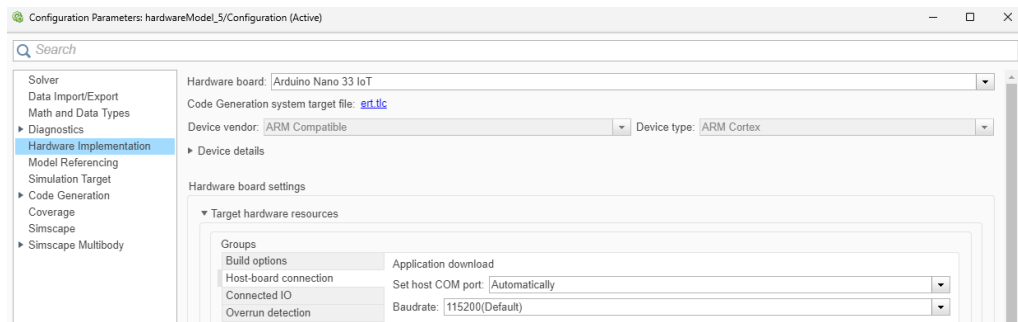
Gambar 5. 15 Test Connection

7. Dan Finish
8. Buka hardware setting



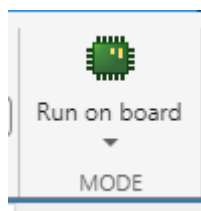
Gambar 5. 16 Hardware settings

9. Buka hardware setting dan sesuaikan konfigurasi seperti berikut



Gambar 5. 17 Konfigurasi hardware

10. Ubah mode menjadi run on board

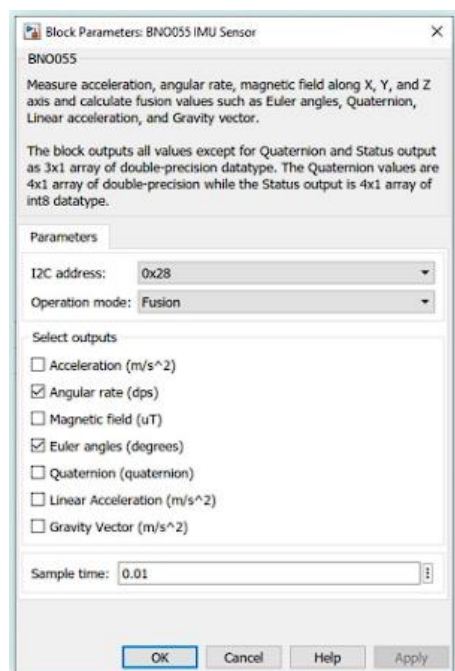


Gambar 5. 18 Konfigurasi hardware

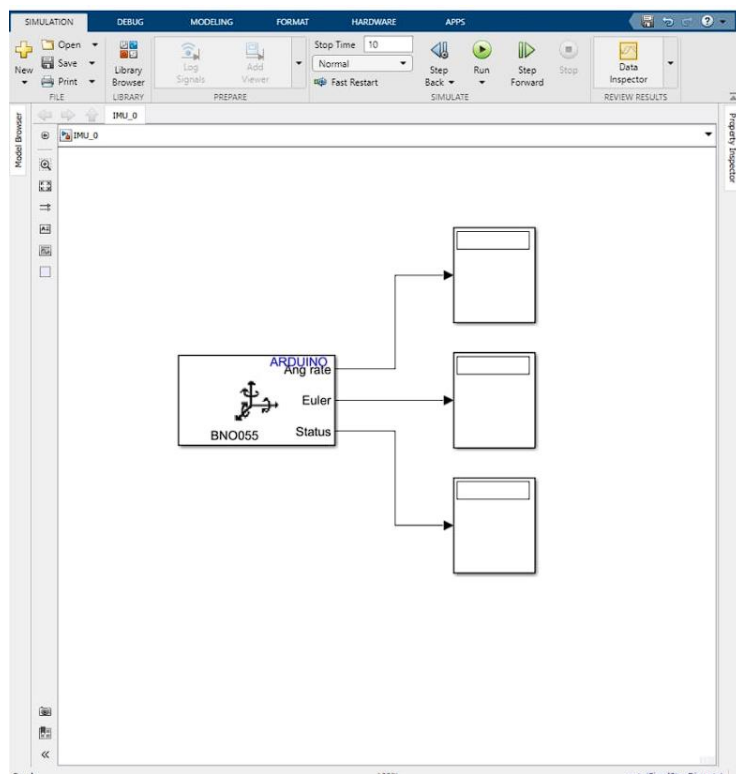
### 5.4.1. Percobaan 1

#### 5.4.2.1. Menjalankan Motorcycle menggunakan Simulink

1. Buatlah File Model dengan nama KelompokXX\_ShiftXX.slx Masukkan di dalam Folder tersebut.
2. Buat rangkaian untuk sensor IMU dengan membuka Library Browser dan ambil komponen berikut:
  - Simulink Support Package for Arduino Hardware > Sensors > BNO055
  - Simulink > Sinks > Display (3x)
  - Pada block BNO055 ubah pengaturan seperti berikut dan hubungkan antar block nya



Gambar 5. 19 Pengaturan BNO055

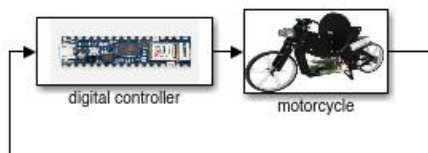


Gambar 5. 20 Block IMU

3. Running dengan klik “Monitor & Tune” pada tab Hardware lihat perubahannya.

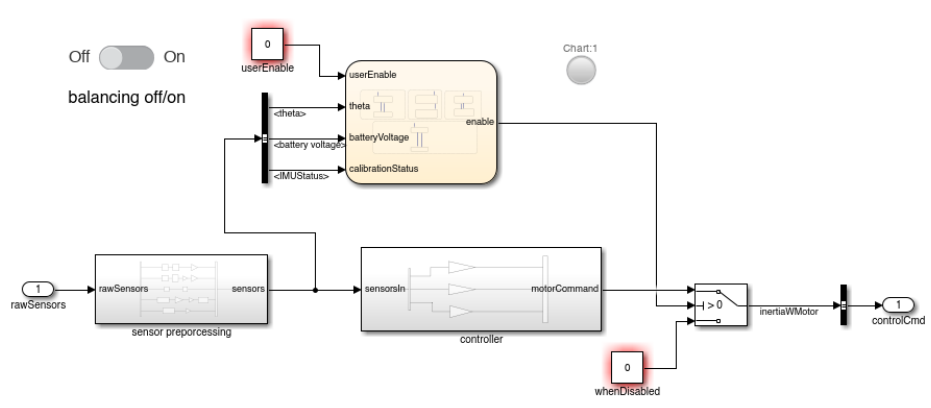
#### 5.4.2. Percobaan 2

1. Buka Modul “hardwareModel\_5.slx” pada folder:  
“..\Arduino\_Engineering\_Kit\_Project\_Files\_Rev\_2\Motorcycle\exercise\_6\_3”
2. Double klik pada “digital controller”



Gambar 5. 21 hardwareModel\_5

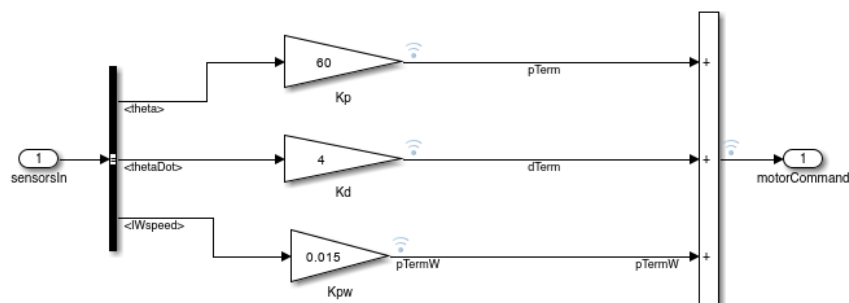
3. Setelah terbuka pastikan tampilannya seperti pada gambar di bawah.



Gambar 5. 22 digital controller hardwareModel\_5

Block digital controller ini adalah block yang berisi kontrol terhadap aktuator dengan input dari feedback yang diberikan oleh sensor.

4. Kemudian buka block “controller” dan pastikan terdapat logo sinyal berwarna biru yang menandakan komponen telah terhubung ke jaringan. Seperti pada gambar berikut.

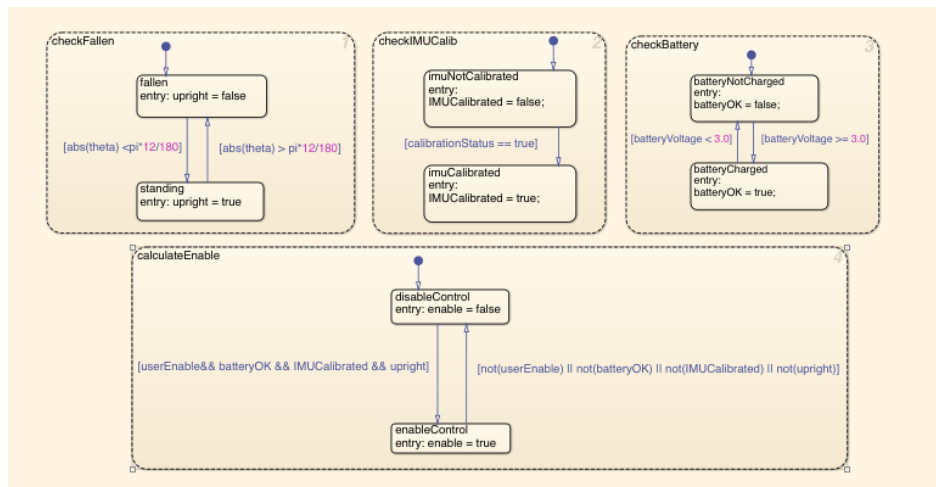


Gambar 5. 23 Block Controller

Block controller ini adalah sistem kontrol yang akan memproses data aktual dari sensor dengan setpoint yang telah diberikan.

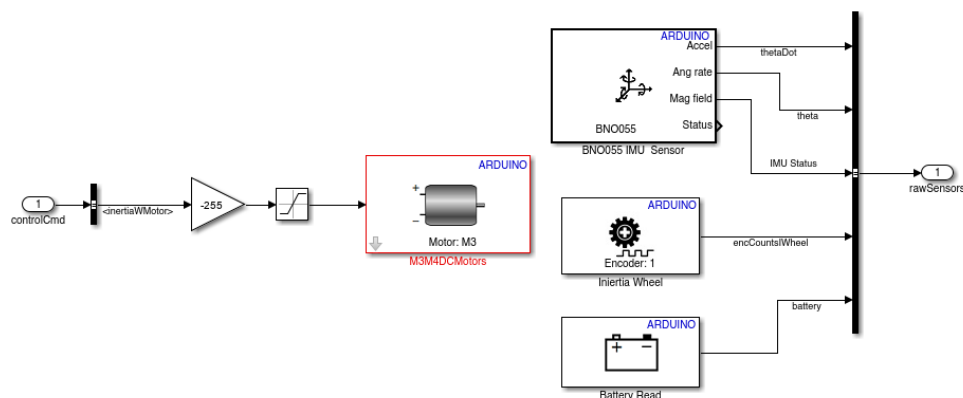
5. Kemudian kembali ke “digital controller” dan buka block chart lalu amati proses kode dari IMU seperti pada gambar berikut.





Gambar 5. 24 Chart code

6. Kemudian kembali ke Modul “hardwareModel\_5.slx” dan masuk ke “motorcycle” sehingga tampilan seperti pada gambar berikut.



Gambar 5. 25 Block "motorcycle"

Block motorcycle ini adalah block yang membaca kondisi aktual dari tiap sensor untuk di-*feedback*-kan ke digital controller, dan juga menggerakkan aktuator sesuai dengan hasil pemrosesan digital controller.

7. Pada terminal Matlab tuliskan kode berikut

```
Ts = 0.01;
```

8. Kemudian kembali ke block “digital controller” dan running dengan klik “Monitor & Tune” pada tab Hardware.
9. Jika terjadi error ulangi langkah konfigurasi simulink

10. Jika berhasil ubah “balancing off/on” menjadi “on” kemudian lakukanlah kalibrasi seperti pada Percobaan 1 hingga block “chart:1” berwarna hijau ketika motorcycle ditegakkan.
11. Motor akan berdiri dengan stabil. Jangan lupa foto motor yang stabil dari atas, samping kanan, samping kiri, belakang, dan depan (perlihatkan motor berdiri dengan stabil).