

## Enhance the Balance of Quadruped Robot using CMPS12

Nasrulloh Azhar<sup>a,1</sup>, Purno Tri Aji<sup>a,2,\*</sup>

<sup>a</sup> Department of Electrical and Electronic Engineering, Vocational Faculty, UNY

<sup>1</sup> [nasrullohazhar.2019@student.uny.ac.id](mailto:nasrullohazhar.2019@student.uny.ac.id); <sup>2</sup> [purno.tri@uny.ac.id](mailto:purno.tri@uny.ac.id)

\* Corresponding Author

### ARTICLE INFO

#### Article History

Received 22 Oct. 2023

Revised 25 Nov. 2023

Accepted 05 Jan. 2024

#### Keywords

Quadruped Robot,

CMPS12 Sensor,

Performance,

Balancing.

### ABSTRACT

Quadruped is a robot that can move stably and flexibly on various types of surfaces. However, when quadruped encounters an uneven surface, it needs a good navigation system to get through it. The analysis of increasing the performance of the quadruped robot is based on the addition of the CMPS12 sensor for navigation. The CMPS12 sensor is used to measure the direction of the robot's orientation. Tests were carried out on four types of obstacles, namely broken road obstacles, sloping road obstacles, rocky road obstacles, and muddy road obstacles. The results of testing the robot on broken road obstacles obtained a maximum slope for the pitch axis of 13° forward, -21° to the rear and for the roll axis at a slope of 24° to the left and -19° to the right. On inclined road obstacles, the robot can pass through obstacles with an average travel time of 8.07 seconds with a maximum slope of 25° on the pitch axis. Then, on the rocky road obstacle, the robot can pass the obstacle with an average travel time of 8.13 seconds, with a maximum slope of 9° on the pitch axis and 8° on the roll axis. Then, on a muddy road obstacle, the robot can pass the obstacle with an average travel time of 11.67 seconds, with a maximum slope of 15° on the pitch axis and -6° on the roll axis.

*Quadruped* merupakan robot yang memiliki kemampuan dapat bergerak secara stabil dan fleksibel di berbagai jenis permukaan. Tetapi ketika menemukan permukaan yang tidak merata, diperlukan sistem navigasi baik untuk dapat melewatinya. Analisis peningkatan performa robot *quadruped* di dasarkan pada penambahan sensor CMPS12 sebagai navigasi. Sensor CMPS12 digunakan untuk mengukur arah orientasi robot. Pengujian dilakukan pada empat jenis rintangan, yaitu rintangan jalan pecah, rintangan jalan miring, rintangan jalan berbatu, dan rintangan jalan berlumpur. Hasil pengujian robot pada rintangan jalan pecah memperoleh kemiringan maksimum pada sumbu pitch sebesar 13° ke depan, -21° ke belakang, dan untuk sumbu roll pada kemiringan 24° ke kiri dan -19° ke kanan. Pada rintangan jalan miring, robot dapat melewati rintangan dengan rata rata waktu tempuh 8,07 detik dengan maksimum kemiringan 25° pada sumbu *pitch*. Kemudian pada rintangan jalan berbatu, robot dapat melewati rintangan dengan rata rata waktu tempuh 8,13 detik, dengan maksimum kemiringan 9° pada sumbu *pitch* 8° pada sumbu roll. Lalu pada rintangan jalan berlumpur, robot dapat melewati rintangan dengan rata rata waktu tempuh 11,67 detik, dengan maksimum kemiringan 15° pada sumbu *pitch* dan -6° pada sumbu *roll*.

This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## 1. Pendahuluan

Pemerintah menggelar Kontes Robot Indonesia (KRI) dalam upaya untuk mengembangkan dan memperkuat daya kreativitas mahasiswa di perguruan tinggi serta dapat menerapkan pengetahuan dan teknologi dalam kehidupan nyata dan meningkatkan kompetensi mahasiswa dalam bidang teknologi terutama robotika.

KRI juga menjadi kesempatan bagi mahasiswa untuk berkolaborasi dan bertukar ide dengan sesama yang memiliki minat dan keahlian serupa. Dalam proses persiapan dan pelaksanaan kontes, kolaborasi ini dapat memperluas wawasan serta memperdalam pemahaman mereka dalam menghadapi berbagai tantangan teknologi masa kini. Diharapkan, melalui partisipasi dalam Kontes Robot Indonesia, mahasiswa tidak hanya mendapatkan pengalaman berharga dalam pengembangan robotika, tetapi juga membangun jaringan yang kuat dan mendukung di dalam dunia teknologi yang terus berkembang pesat (Kemdikbud, 2023).

KRI merupakan acara tahunan yang mempertandingkan kemampuan dan kreativitas mahasiswa dalam merancang dan mengembangkan robotika. Kompetisi ini terbuka untuk semua mahasiswa di seluruh perguruan tinggi Indonesia dan meliputi berbagai instansi pemerintahan seperti kementerian, lembaga, dan dinas, yang tercatat dalam Pangkalan Data Pendidikan Tinggi (PDDikti). Kompetisi Reka Inovasi (KRI) diprakarsai oleh lembaga Balai Pengembangan Talenta Indonesia (BPTI) yang merupakan bagian dari Pusat Prestasi Nasional (Puspresnas) Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi (Kemendikbudristek) Republik Indonesia. Kompetisi Rekayasa Indonesia (KRI) digelar pertama kali pada tahun 2003 di bawah naungan Direktorat Jenderal Pendidikan Tinggi, Departemen Pendidikan dan Kebudayaan pada era tersebut. Saat ini, Kontes Robot Indonesia mengadakan pertandingan di 7 (tujuh) kategori, yakni: Kontes Robot ABU Indonesia (KRAI); Kontes Robot SAR Indonesia (KRSRI); Kontes Robot Sepak Bola Indonesia (KRSBI) dengan roda; Kontes Robot Sepak Bola Indonesia (KRSBI) humanoid; Kontes Robot Seni Tari Indonesia (KRSTI); Kontes Robot Tematik Indonesia (KRTMI); serta Kontes Robot Bawah Air Indonesia (KRBAI). (KRI, 2023).

KRSRI adalah acara yang menggantikan Kontes Robot Pemadam Api (KRPAI) di Indonesia. KR-PAI memiliki fokus utama pada upaya menemukan dan menyelamatkan korban bencana yang sering terjadi di Indonesia. Pilihan tema untuk tahun 2023 dipengaruhi oleh tantangan yang muncul setelah gempa melanda Cianjur pada akhir November 2022. Robot berjalan sendiri yang ada di KRSRI 2023 tidak dapat langsung digunakan di lokasi setelah bencana. Tetapi, KRSRI 2023 diharapkan bisa meningkatkan kemampuan anak bangsa dalam menguasai robot otonom berkaki setelah terjadi bencana. Maka, jika suatu saat nanti tercipta robot yang memiliki ukuran dan kekuatan yang lebih besar, pengendaliannya akan menjadi lebih mudah dan lebih cepat karena seorang individu muda telah mendapatkan pelatihan dalam menggunakan model miniatur dan gambar ilustrasi yang menggambarkan situasi setelah terjadinya bencana di arena KRSRI 2023 (Puspresnas, 2023).

Tugas robot pada KRSRI tahun 2023 berbeda dengan tugas robot KRPAI dan KRSRI sebelumnya. Sebelumnya, robot harus meniuip lilin, tetapi kali ini robot tidak harus memadamkan api. Fokusnya adalah melewati berbagai macam variasi rintangan yang menggambarkan kondisi pasca bencana, terutama setelah terjadi gempa. Tantangan bagi robot adalah untuk mengamankan orang-orang yang terjebak di lokasi tertentu dan mengantarkannya ke zona yang aman, yang telah ditetapkan sebelumnya. Pada tahun 2023, terdapat empat macam tantangan yang berbeda dalam kontes divisi ini jika dibandingkan dengan tahun-tahun sebelumnya, yaitu jalan miring, jalan pecah, jalan berbatu, dan jalan berlumpur. Untuk menggambarkan situasi setelah terjadi bencana, terutama gempa. Agar peserta dapat dengan lebih efisien dan cepat membuat rintangan, digunakanlah batu koral putih ukuran 3-5 cm yang biasanya digunakan dalam taman untuk membangun rintangan jalan berbatu. Sedangkan jalanan yang berlumpur diterapkan dengan menggunakan kelereng sebagai rintangan bagi kaki robot agar dapat terjebak di tumpukan kelereng. Membuat jalan pecah membutuhkan upaya dengan cara memotong papan triplek berbentuk pola yang telah ditetapkan. Jalan miring terdiri dari bagian yang menurun dan bagian yang menaik. Di jalan yang naik,

terdapat anak tangga yang diberikan agar kaki robot mempunyai pijakan. Selain halangan yang disebutkan, batas tinggi dinding pada lintasan hanya 10 cm dari atas lantai (Puspresnas, 2023).

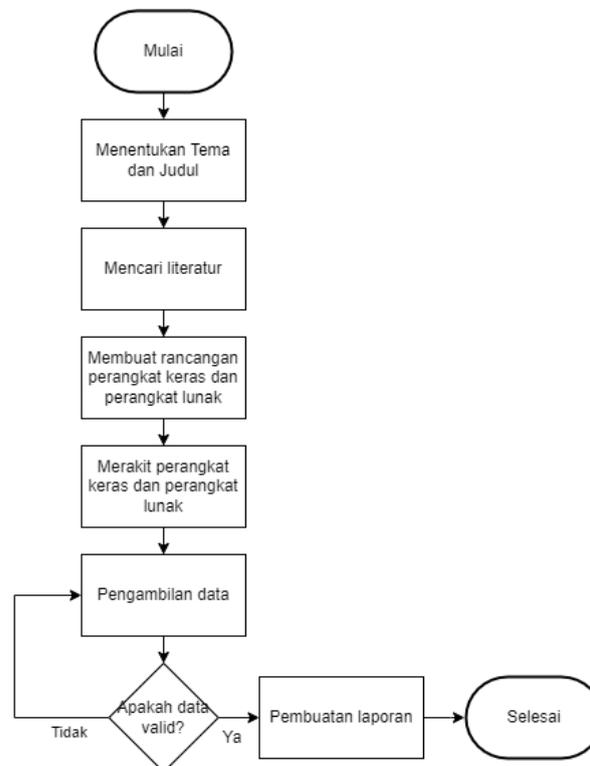
KRSRI 2023 diikuti oleh 93 tim dari 93 perguruan tinggi yang ada di Indonesia. Berdasarkan data hasil skor wilayah 1 dan 2, tercatat hanya ada 44 tim atau 47.22% dari seluruh tim yang mampu melewati rintangan dari jalan pecah hingga jalan berlumpur.

Pada formasi tim, penulis berada pada posisi *programmer* yang bertugas mengatur jalannya robot meliputi: algoritma gerak, memilih sensor yang akan digunakan pada robot, dan strategi untuk memperoleh point pada perlombaan. Berdasarkan rintangan dan data tersebut, robot kemudian disesuaikan sedemikian rupa agar dapat melewati rintangan dengan baik. Salah satu caranya adalah dengan menambahkan sensor CMPS12 yang dapat membaca sudut rotasi seperti *yaw*, *pitch*, *roll* agar robot dapat mengetahui dimana posisinya saat ini.

## 2. Metode Penelitian

### 2.1. Prosedur Penelitian

Prosedur penelitian dilaksanakan sesuai diagram alir yang telah disusun menjadi sebuah diagram blok pada gambar dibawah ini:



**Fig. 1.** Diagram alir penelitian

Secara garis besar, langkah penelitian digambarkan dalam diagram blok seperti pada Fig 1. Langkah awal dalam penelitian ini adalah menentukan tema yaitu tentang robot berkaki menggunakan sensor IMU. Langkah selanjutnya yaitu mengumpulkan literatur yang terkait dengan penelitian ini. Kemudian, dilanjutkan dengan membuat rancangan perangkat keras dan perangkat lunak yang akan dibuat. Jika perancangan sudah dilakukan, langkah selanjutnya yaitu merakit atau membuat perangkat keras dan perangkat lunak yang telah dibuat desainnya sebelumnya. Setelah perangkat sudah dibuat, dilakukan pengujian data dan pengambilan data. Ketika *output* data yang diamati tidak valid, maka akan dilakukan pengujian ulang untuk pengambilan datanya. Setelah data yang diinginkan diperoleh dari serangkaian pengujian, langkah selanjutnya adalah membuat laporan untuk menganalisis dan melaporkan hasil kinerja alat yang telah dibuat.

## 2.2. Analisis Kebutuhan

### STM32F407G Discovery

STM32F407VGT6 (STM32F407G Discovery) merupakan sebuah mikrokontroler berbasis 32-bit dan menggunakan arsitektur RISC (*Reduced Instruction Set Computer*) yang dapat beroperasi hingga frekuensi 168 MHz. Seri STM32F407xx memiliki (*flash Memory* hingga 1 Mbyte, 192 Kbytes SRAM), dan 4 Kbytes SRAM cadangan. Berbagai macam I/O dan periferal yang ditingkatkan terhubung ke dua bus APB, tiga bus AHB, dan matriks bus multi-AHB 32-bit. Seri STM32F407xx memiliki tiga ADC 12-bit, dua DAC, RTC *low power*, dua belas timer 16-bit tipe umum termasuk dua timer PWM untuk kendali motor, dan dua timer 32-bit tipe umum. Dilengkapi juga dengan *Random Number Generator* (RNG).

### STM32 Nucleo L432KC

STM32L432KC adalah mikrokontroler ultra-low-power berbasis high-performance Arm Cortex-M4 32-bit RISC (*Reduced Instruction Set Computer*) yang dapat beroperasi hingga frekuensi 80 MHz. STM32L432KC menggunakan memori berkecepatan tinggi (*flash memory* hingga 256 Kbyte, 64 Kbyte SRAM), *Quad SPI flash memory*, dan berbagai macam I/O, juga periferal yang terhubung ke dua bus APB, dua bus AHB, dan matriks bus multi-AHB 32-bit.

### Servo Dynamixel AX-12A

Dynamixel AX-12A merupakan motor servo yang memiliki torsi hingga 12 kgf.cm dan juga dilengkapi kemampuan *networking* melalui UART TTL half duplex multidrop. Dynamixel AX-12A terdiri dari *reduction gear*, *precision DC Motor*, dan rangkaian kontrol yang telah dilengkapi dengan kemampuan komunikasi data. AX-12A merupakan versi terbaru dari AX-12+ dengan kemampuan yang sama tetapi dengan penambahan desain eksternal.

### Sensor CMPS12

CMPS12 merupakan modul sensor kompas + IMU (*Inertial Measurement Unit*) menggunakan 3 sumbu magnetometer, gyro, dan accelerometer. Modul CMPS12 ditenagai chip sensor 9-axis IMU BNO055 yang diproduksi oleh Bosch. Modul ini bekerja pada rentang catu daya fleksibel antara 3.3V - 5V. Modul ini mendukung dua tipe komunikasi data, yaitu I2C dan Serial.

### Baterai Lipo 2200mAh

Baterai Gens ace G-Tech Soaring 2200mAh 11.1V 30C 3S1P Lipo Battery Pack with XT60 plug adalah baterai isi ulang berjenis polimer lithium (LiPo). Baterai ini memiliki kapasitas 2200mA dengan tegangan 3S (11.1V) dan arus pengosongan puncak 30C. Baterai ini dilengkapi dengan konektor XT60. Baterai ini menggunakan teknologi pengisian otomatis yang stabil untuk memungkinkan kapasitas sel tunggal mencapai 1300mAh. Teknologi ini menghasilkan baterai yang lebih ringan dan lebih efisien. Baterai ini cocok untuk model pesawat RC berukuran sedang hingga besar dan perangkat robotika. Baterai ini dapat memberikan tenaga yang kuat dan tahan lama untuk model-model tersebut.

### STM32CubeMX

STM32CubeMX adalah *software* berbasis GUI (*Graphical User Interface*) buatan STMicroelectronics untuk mempermudah konfigurasi mikrokontroler dan mikroprosesor STM32. Langkah awal yang dapat dilakukan oleh pengguna adalah memilih jenis mikrokontroler atau mikroprosesor STM32 yang diinginkan. Kemudian pengguna dapat melakukan konfigurasi seperti GPIO, clock, dan beberapa *peripheral* seperti *timer*, analog, dan sebagainya. Pengembangan *software*

dan *middleware* dapat diperluas dengan menggunakan Paket Ekspansi STM32Cube yang lebih canggih. Paket ini dapat diunduh dari STMicroelectronics langsung melalui STM32CubeMX, atau diinstal dari penyimpanan lokal. Setelah semua inisiasi yang diinginkan selesai, pengguna dapat melakukan finalisasi pada menu Project Manager lalu bisa Generate Code. Kode yang sudah dibuat dapat langsung digunakan atau dibuka melalui IDE yang dipilih sebelumnya.

### STM32 Cube IDE

STM32CubeIDE merupakan sebuah platform IDE berbasis C/C++ yang memiliki fungsi untuk mengatur periferal, pembuatan kode, kompilasi kode, dan fitur debugging untuk mikrokontroler serta mikroprosesor STM32. Platform ini berbasis Eclipse/CDT dan rangkaian alat GCC untuk pengembangan, serta GDB untuk debugging. Dengan STM32CubeIDE, pengguna dapat mengintegrasikan berbagai *plugin* yang sudah ada untuk melengkapi fitur dari IDE Eclipse. Konfigurasi dan pembuatan *project* dari STM32CubeMX dapat digabungkan dengan CubeIDE, sehingga memberikan pengalaman sehingga menghemat waktu instalasi dan pengembangan. Setelah memilih mikrokontroler atau mikroprosesor STM32 yang kosong atau yang sudah terkonfigurasi sebelumnya, *project* akan langsung dibuat dan kode inisialisasi akan dihasilkan. Pengguna fleksibel untuk kembali ke tahap inisialisasi dan konfigurasi periferal serta membuat ulang kode inisialisasi tanpa mempengaruhi kode yang sudah ada. STM32CubeIDE juga memiliki alat analisis build dan stack untuk memberikan informasi yang berguna mengenai status proyek dan kebutuhan memori yang diperlukan. Tidak hanya itu, pada *software* STM32CubeIDE juga menyediakan berbagai fitur debug, tampilan register inti CPU, memori, dan register periferal, serta pemantauan variabel secara langsung.

### KiCad

KiCad merupakan *software open-source* yang digunakan untuk keperluan *Electronic Design Automation* (EDA). *Software* ini memfasilitasi pembuatan skema dan penataan PCB dengan menghasilkan *output* berformat Gerber. *Software* ini dapat dijalankan pada Windows, Linux, dan macOS. *Software* ini dilisensikan di bawah GNU GPL v3.

## 3. Perancangan Alat

Perancangan alat meliputi pembuatan desain rangkaian elektronik, mekanik, dan program yang akan dibuat. Berikut beberapa tahapan perancangan alat yang dilakukan:

### 3.1. Rangkaian Elektronik

Proses desain rangkaian elektronik pada alat ini menggunakan *software KiCad*. Berikut beberapa desain elektronik yang ada pada alat ini,

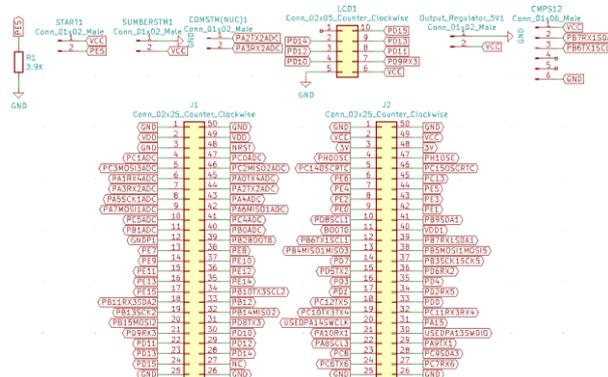


Fig. 2. Skematik rangkaian *shield* STM32F407VGT6

Desain rangkaian *shield* STM32F407VGT6 pada gambar 2 berisi 4 konektor XH 2.54 isi 2 yang masing masing untuk koneksi ke *push button*, VCC STM32, komunikasi ke STM32L432KC, dan ke Regulator. 1 *female header* isi 6 untuk CMPS12. 1 *male header* isi 10 untuk LCD.

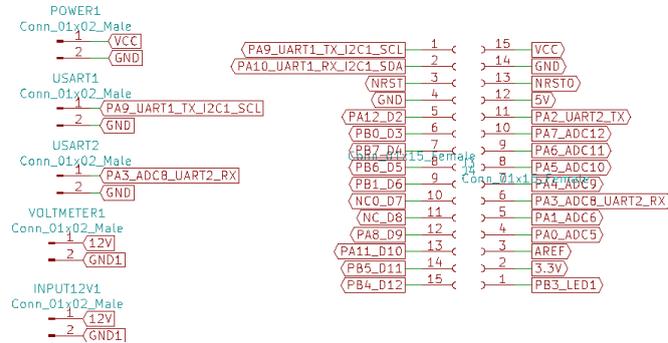


Fig. 3. Skematik rangkaian *shield* STM32L432KC

Desain rangkaian *shield* STM32L432KC pada gambar 3 berisi 5 buah *socket* XH 2.54, 3 buah koneksi masing masing untuk VCC, komunikasi ke STM32F407VGT dan ke Hub Servo, 2 buah *socket* sisanya untuk slot cadangan.

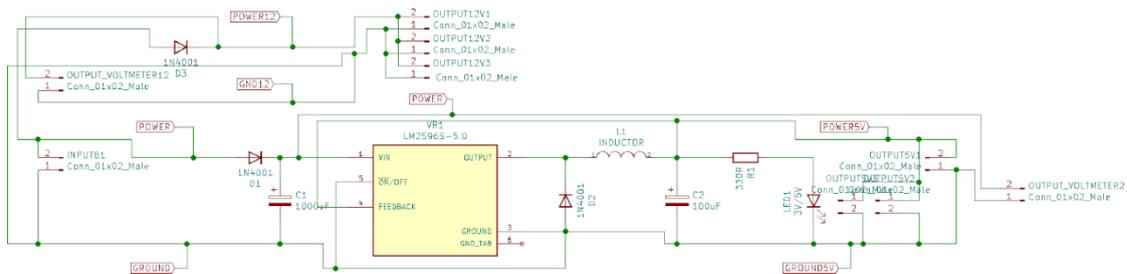


Fig. 4. Skematik rangkaian *regulator* 5V DC

Desain dari rangkaian *regulator* pada gambar 4 menggunakan beberapa komponen, diantaranya 3 buah dioda 1N4001, 1 buah kapasitor 1000uF, 1 buah kapasitor 100uF, 1 buah LED, 1 buah IC LM2596, 1 buah induktor 100uH, dan 1 buah resistor 330.

### 3.2. Rangkaian Mekanik

Desain rangkaian mekanik meliputi *body* robot yang digunakan. Berikut beberapa desain robot dari beberapa arah pandangan

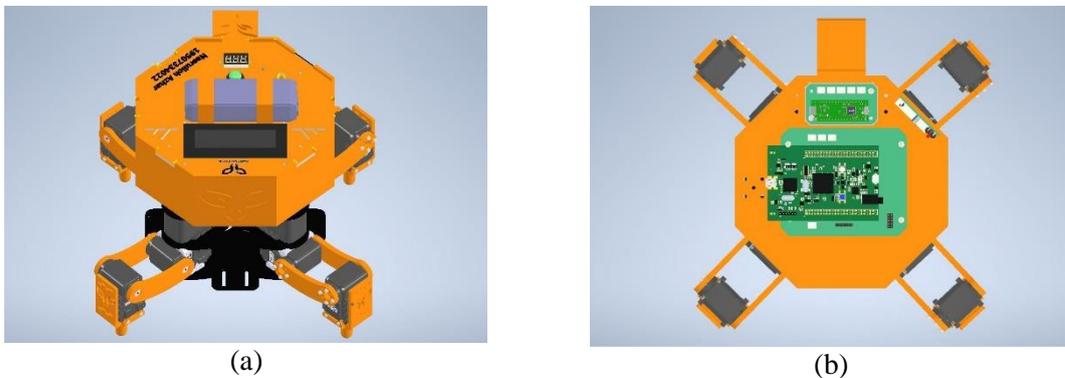
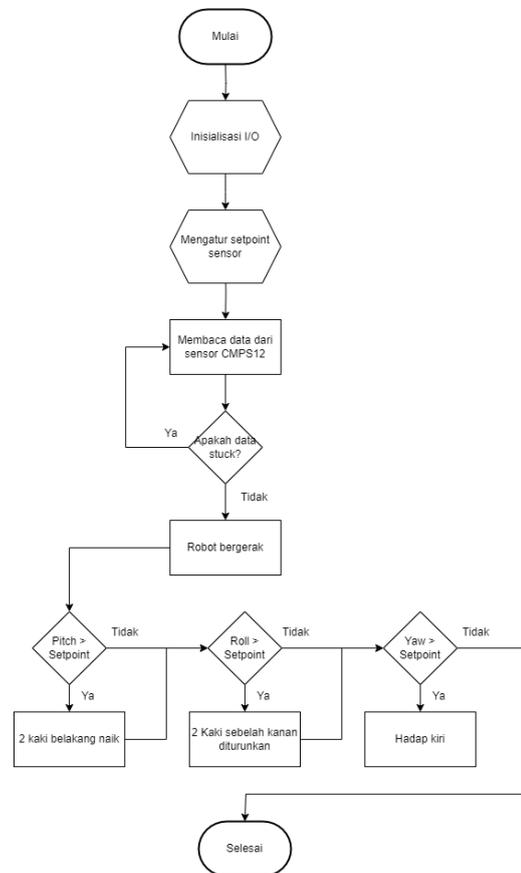


Fig. 5. (a) Robot tampak depan (b) tampak atas dan posisi penempatan elektronik

Rangkaian program meliputi *flowchart* algoritma pergerakan robot, penentuan *pinout* yang digunakan pada STM32F407VGT dan STM32L432KC. Berikut beberapa rangkaian program yang dibuat.



**Fig. 6.** Flowchart algoritma pergerakan robot

Pada gambar 7 dijelaskan cara kerja alat yang akan digunakan. Mulai dari inisialisasi *input / output*, kemudian mengatur *set point* dari sensor yang selanjutnya akan dijadikan acuan sebagai titik utama yang ingin dicapai dalam kondisi stabil. Kemudian setelah itu robot dinyalakan dan mikrokontroler membaca data dari sensor CMPS12 yang telah terpasang di *body* robot. Ketika terjadi *stuck* pada sensor atau data yang terbaca itu *error*, maka proses pengambilan data akan diulang dari awal saat menyalakan robot. Selanjutnya, ketika ditemukan ada data dari sensor yang melebihi atau kurang dari setpoint yang ditetapkan, maka robot akan melakukan gerakan yang sesuai agar robot bisa tetap stabil saat berjalan.

## 4. Hasil dan Pembahasan

### 4.1. Implementasi Alat

Implementasi alat dari perencanaan yang sudah direncanakan sebelumnya dijelaskan dalam bagian dibawah ini.

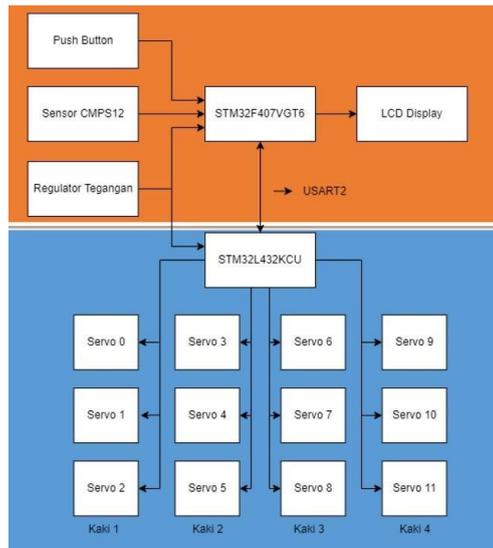


Fig. 7. Diagram blok sistem kerja alat

Secara garis besar, sistem alat yang dibuat dibagi menjadi 2 bagian yaitu bagian Kepala dan Kaki. Pada gambar 8, bagian kepala ditandai dengan warna oranye yang berisi kontrol utama robot seperti mikrokontroler STM32F407VGT, *Push button*, sensor CMPS12, *lcd display*, serta regulator tegangan. Sedangkan bagian kaki ditandai dengan warna biru yang berisi kontrol utama gerak kaki robot, berisi STM32L432KC untuk mengontrol jalannya 12 servo Dynamixel AX-12A.

#### 4.2. Hasil Pengujian Jalan Pecah

Pada pengujian ini dilakukan pengambilan data pada sudut orientasi *pitch* dan *roll* sembari melihat respon aksi robot. Dibawah ini merupakan contoh dari salah satu data yang diamati melalui *Debug STM32CubeIDE*.

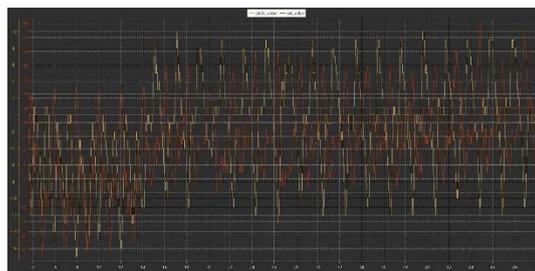


Fig. 8. Grafik pengujian sumbu *pitch* dan *roll* secara *real* pada rintangan jalan pecah

Gambar 9 merupakan hasil pengujian sensor pada robot yang diamati melalui debug via Serial Wire Viewer (SWV) di aplikasi STM32CubeIDE.

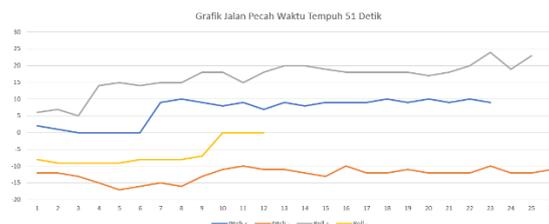


Fig. 9. Grafik pengujian sumbu *pitch* dan *roll* pada rintangan jalan pecah dengan waktu tempuh 51 detik



**Fig. 10.** Grafik pengujian sumbu *pitch* dan *roll* pada rintangan jalan pecah dengan waktu tempuh 42,8 detik

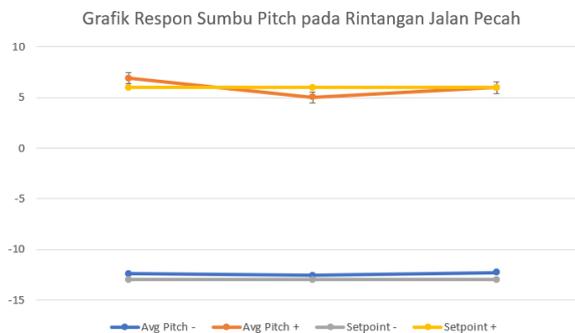


**Fig. 11.** Grafik pengujian sumbu *pitch* dan *roll* pada rintangan jalan pecah dengan waktu tempuh 42,5 detik

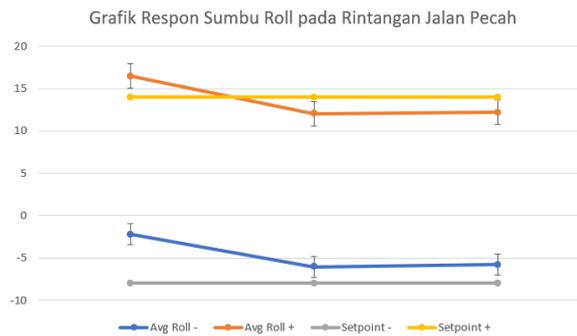
Setelah dilakukan pengujian terhadap rintangan jalan pecah, robot dapat melalui rintangan jalan pecah dengan waktu tempuh terlama 51 detik, dan tercepat di angka 42,5 detik dengan rata rata waktu tempuh 45,43 detik. Kemudian, dengan menentukan *set point* yang bertujuan untuk membatasi pergerakan robot berdasarkan jenis kemiringannya yang dibagi menjadi sumbu *pitch* dan *roll* masing masing positif dan negatif, didapatkan hasil seperti pada gambar 4.23 hingga 4.25. Berdasarkan data yang diperoleh dengan setpoint sumbu *pitch* sebesar -13 ke belakang dan 6 kedepan, percobaan pertama memperoleh waktu tempuh sebesar 51 detik dengan titik maksimum sumbu *pitch* (+) berada pada titik 10, pada sumbu *pitch* (-) pada titik -17, pada sumbu *roll* (+) pada titik 24, dan pada sumbu *roll* (-) berada pada titik -9.

Pada percobaan berikutnya, robot dapat melalui rintangan dengan waktu tempuh 42,8 detik, dengan titik maksimum sumbu *pitch* (+) berada pada titik 13, pada sumbu *pitch* (-) pada titik -21, pada sumbu *roll* (+) pada titik 23, dan pada sumbu *roll* (-) berada pada titik -19.

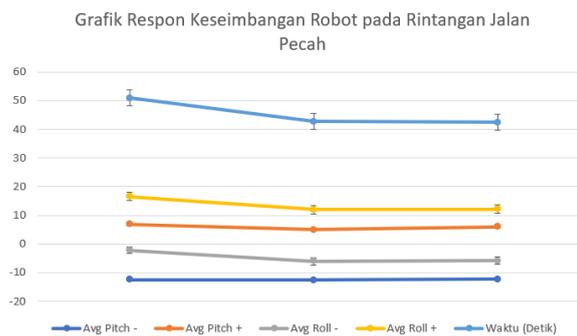
Pada percobaan terakhir, robot dapat melalui rintangan dengan waktu tempuh 42,5 detik, dengan titik maksimum sumbu *pitch* (+) berada pada titik 13, pada sumbu *pitch* (-) pada titik -21, pada sumbu *roll* (+) pada titik 23, dan pada sumbu *roll* (-) berada pada titik -19.



**Fig. 12.** Grafik respon sensor terhadap rintangan jalan pecah pada sumbu *pitch*



**Fig. 13.** Grafik respon sensor terhadap rintangan jalan pecah pada sumbu *roll*

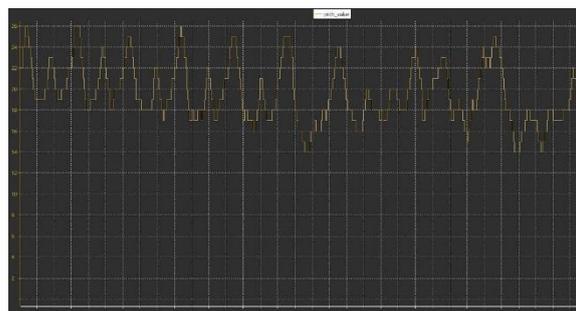


**Fig. 14.** Grafik respon kestabilan terhadap rintangan jalan pecah

Gambar 4.26 sampai 4.28 merupakan grafik data respon sensor terhadap rintangan jalan pecah, untuk respon sumbu *pitch* jauh lebih baik dibandingkan sumbu *roll*, ini dikarenakan kebanyakan posisi pecahan berada pada sisi kanan dan kiri robot yang ketika terinjak oleh kaki robot menyebabkan robot miring ke kanan dan ke kiri yang merupakan sumbu *roll*.

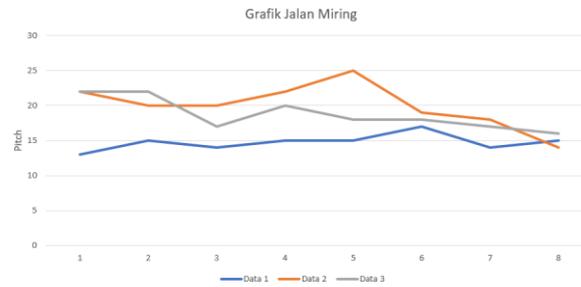
### 4.3. Hasil Pengujian Jalan Miring

Pada pengujian ini dilakukan pengambilan data pada sudut orientasi *pitch* sembari melihat respon aksi robot. Dibawah ini merupakan contoh dari salah satu data yang diamati melalui *Debug* STM32CubeIDE.



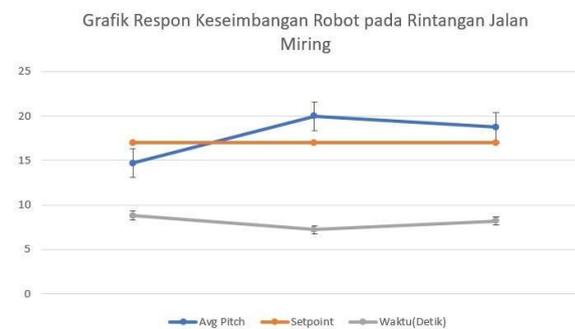
**Fig. 15.** Grafik pengujian sumbu *pitch* secara real pada rintangan jalan miring

Gambar 16 merupakan hasil pengujian sensor pada robot yang diamati melalui debug via Serial Wire Viewer (SWV) di aplikasi STM32CubeIDE.



**Fig. 16.** Grafik pengujian sumbu pitch pada jalan miring

Gambar 17 merupakan hasil data dari pengamatan robot pada rintangan jalan miring. Tercatat dengan waktu tempuh terlama 8,8 detik dan tercepat 7,2 detik dengan rata rata waktu tempuh 8,07 detik. Robot cenderung condong ke depan dilihat dari data dengan angka maksimum pada kemiringan 25°, tetapi dapat dengan stabil melewati turunan tanpa tergelincir.

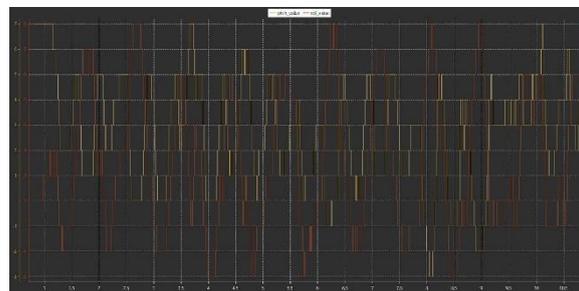


**Fig. 17.** Grafik respon sensor terhadap rintangan jalan miring

Gambar 18 merupakan grafik data respon sensor terhadap rintangan jalan miring, untuk respon sumbu *pitch* terlihat terjadi lonjakan untuk condong kedepan dengan waktu yang lebih lama, sedangkan ketika sumbu *pitch* stabil, waktu yang diperlukan lebih sedikit.

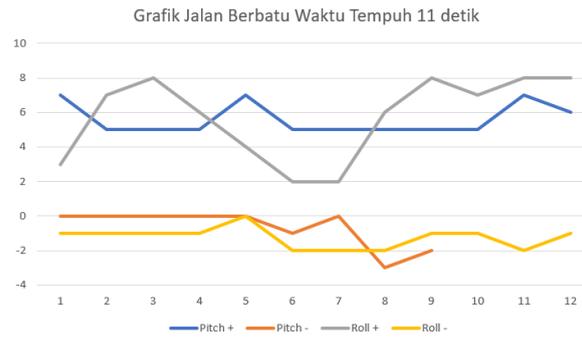
#### 4.4. Hasil Pengujian Rintangan Jalan Berbatu

Pada pengujian ini dilakukan pengambilan data pada sudut orientasi *pitch* dan *roll* sembari melihat respon aksi robot. Dibawah ini merupakan contoh dari salah satu data yang diamati melalui *Debug* STM32CubeIDE.

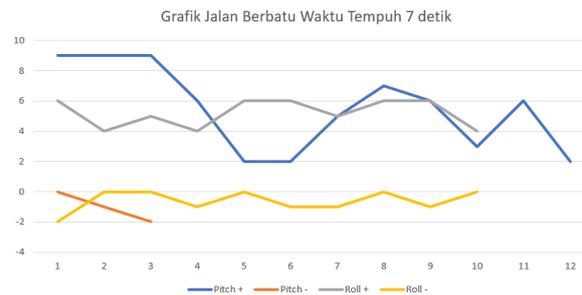


**Fig. 18.** Grafik pengujian sumbu *pitch* dan *roll* secara real pada rintangan jalan berbatu

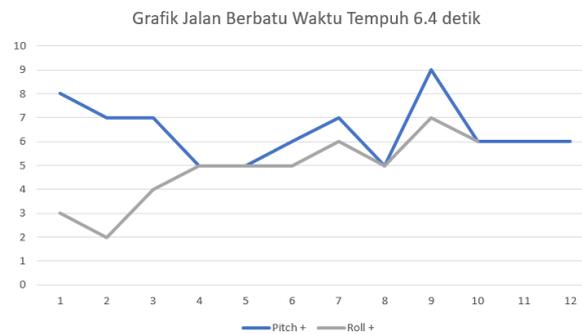
Gambar 19 merupakan grafik data respon sensor terhadap rintangan jalan berbatu, untuk respon sumbu *pitch* dan *roll* terlihat stabil dan tidak mengalami lonjakan yang signifikan yang mengganggu jalannya robot saat melewati rintangan berbatu.



**Fig. 19.** Grafik pengujian sumbu *pitch* dan *roll* pada rintangan jalan berbatu dengan waktu tempuh 11 detik



**Fig. 20.** Grafik pengujian sumbu *pitch* dan *roll* pada rintangan jalan berbatu dengan waktu tempuh 7 detik

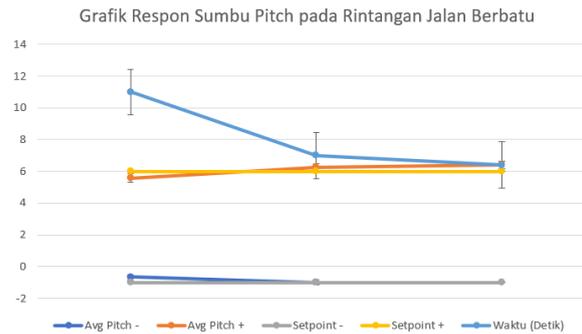


**Fig. 21.** Grafik pengujian sumbu *pitch* dan *roll* pada rintangan jalan berbatu dengan waktu tempuh 6,4 detik

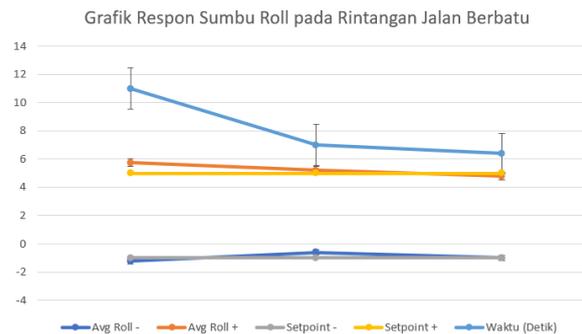
Setelah dilakukan pengujian terhadap rintangan jalan berbatu, robot dapat melalui rintangan jalan berbatu dengan waktu tempuh terlama 11 detik, dan tercepat di angka 6,4 detik dengan rata rata waktu tempuh 8,13 detik. Berdasarkan penentuan *set point*, didapatkan hasil seperti pada Gambar 4.35 hingga 4.37. Terlihat dari data yang diperoleh dengan setpoint sumbu *pitch* sebesar -1 ke belakang dan 6 kedepan, percobaan pertama memperoleh waktu tempuh sebesar 11 detik dengan titik kemiringan depan maksimum sebesar 7 untuk sumbu *pitch*, lalu mengalami kemiringan belakang maksimum sebesar -3.

Pada percobaan berikutnya, robot dapat melalui rintangan dengan waktu tempuh 7 detik, dengan titik maksimum sumbu *pitch* (+) pada awal robot dinyalakan berada pada titik 9, pada sumbu *pitch* (-) pada titik -2, pada sumbu *roll* (+) dan *roll* (-) cenderung lebih stabil.

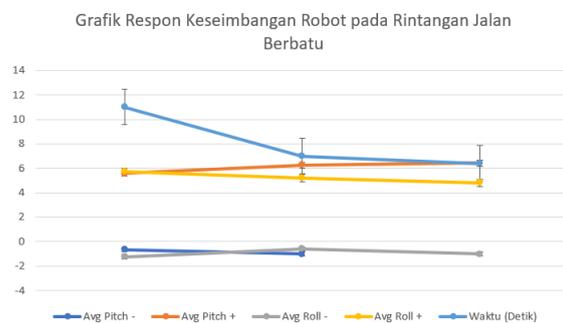
Pada percobaan terakhir, robot dapat melalui rintangan dengan waktu tempuh 6,4 detik, dengan titik maksimum sumbu *pitch* (+) berada pada titik 9, pada sumbu *roll* (+) pada titik terendah di 2. Pada percobaan kali ini, tidak terdapat kemiringan ke arah negatif (-) pada kedua sumbu, yang artinya robot tidak mengalami kemiringan ke arah kanan dan belakang.



**Fig. 22.** Grafik respon sensor terhadap rintangan jalan berbatu pada sumbu *pitch*



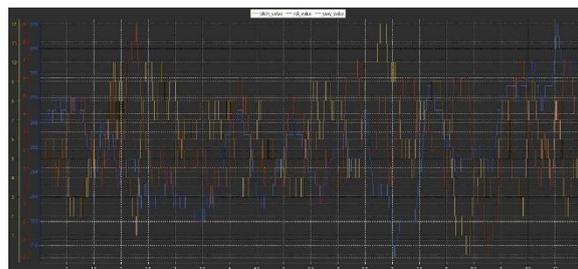
**Fig. 23.** Grafik respon sensor terhadap rintangan jalan berbatu pada sumbu *roll*



**Fig. 24.** Grafik respon kestabilan terhadap rintangan jalan berbatu

#### 4.5. Hasil Pengujian Rintangan Jalan Berlumpur

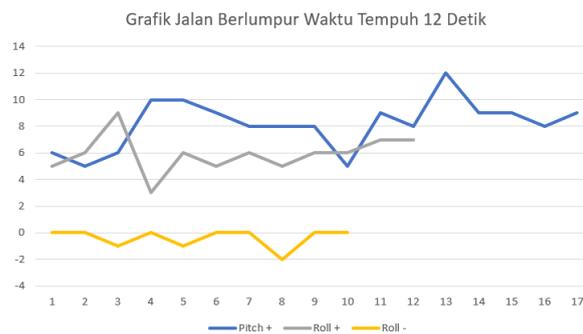
Pada pengujian ini dilakukan pengambilan data pada sudut orientasi *pitch*, *roll*, dan *yaw* sembari melihat respon aksi robot. Dibawah ini merupakan contoh dari salah satu data yang diamati melalui *Debug* STM32CubeIDE.



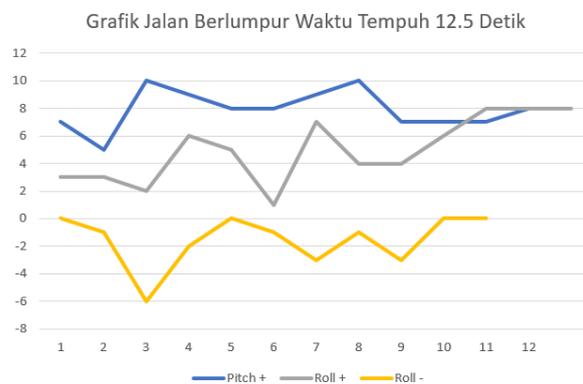
**Fig. 25.** Grafik pengujian sumbu pitch dan roll secara real pada rintangan jalan berlumpur



**Fig. 26.** Grafik pengujian sumbu *pitch* dan *roll* pada rintangan jalan berlumpur dengan waktu tempuh 10,5 detik



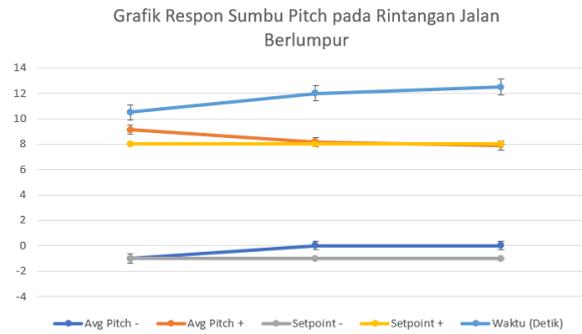
**Fig. 27.** Grafik pengujian sumbu *pitch* dan *roll* pada rintangan jalan berlumpur dengan waktu tempuh 12 detik



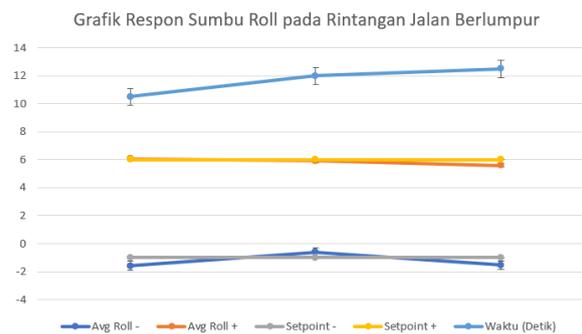
**Fig. 28.** Grafik pengujian sumbu *pitch* dan *roll* pada rintangan jalan berlumpur dengan waktu tempuh 12,5 detik

Setelah dilakukan pengujian terhadap rintangan jalan berlumpur, robot dapat melalui rintangan jalan berlumpur dengan waktu tempuh terlama 12,5 detik, dan tercepat di angka 10,5 detik dengan rata rata waktu tempuh 11,67 detik. Berdasarkan penentuan setpoint, didapatkan hasil seperti pada gambar 4.43 hingga 4.45. Terlihat dari data yang diperoleh dengan *set point* sumbu *pitch* sebesar -1 ke belakang dan 8 kedepan, percobaan pertama memperoleh waktu tempuh sebesar 10,5 detik dengan *point* kemiringan maksimum pada sumbu *pitch* (+) di titik 15.

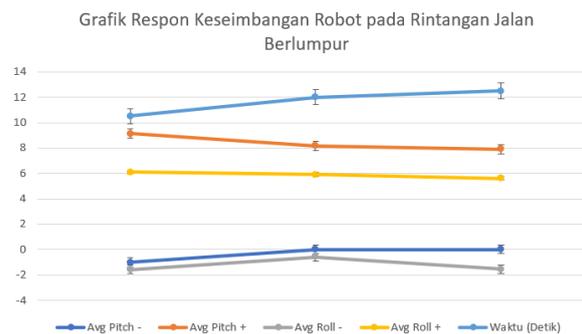
Pada percobaan berikutnya, robot dapat melewati rintangan dengan waktu tempuh 12 detik tanpa mengalami kemiringan ke arah belakang dan hanya sedikit kemiringan ke kanan. Tercatat *point* maksimum untuk kemiringan sumbu *pitch* (+) sebesar 12. Pada percobaan terakhir, robot dapat melewati rintangan dengan waktu tempuh 12,5 detik tanpa mengalami kemiringan ke arah belakang seperti sebelumnya. Tercatat robot mengalami kemiringan maksimum pada sumbu *pitch* (+) pada 10 dan pada sumbu *roll* (-) -6.



**Fig. 29.** Grafik respon sensor terhadap rintangan jalan berlumpur pada sumbu *pitch*



**Fig. 30.** Grafik respon sensor terhadap rintangan jalan berlumpur pada sumbu *roll*



**Fig. 31.** Grafik respon kestabilan terhadap rintangan jalan berlumpur

## 5. Kesimpulan

Hasil pengujian yang dilakukan terhadap robot yang menggunakan dua mikrokontroler berbeda yaitu STM32F407VG untuk kontrol sensor dan STM32L432KC untuk kontrol servo, terbukti bahwa penggunaan pendekatan ini efektif dalam memisahkan fungsi kontroler antara sensor dan servo tanpa mengganggu kinerja masing-masing perangkat yang memiliki spesifikasi yang berbeda. Performa robot dalam melewati berbagai rintangan, seperti jalan pecah, jalan miring, jalan berbatu, dan jalan berlumpur, menunjukkan hasil yang baik. Dengan rata-rata waktu tempuh yang bervariasi untuk setiap jenis rintangan, robot mampu mengatasi kemiringan yang signifikan, seperti kemiringan pada sumbu *pitch* hingga  $25^\circ$ , dengan kemampuan stabilisasi yang baik. Penggunaan sensor tambahan, CMPS12, sebagai navigasi tambahan juga meningkatkan tingkat keberhasilan robot dalam melewati rintangan menjadi 75% dari total percobaan yang diujikan, mencapai 12 kali keberhasilan dari 16 kali percobaan.

## References

- [1] Panitia Pusat Kontes Robot Indonesia, & Kemdikbud. (2023). Kontes Robot Indonesia 2023.

- 
- [2] Kemdikbud, P. (2023). Kontes Robot Indonesia 2023. <https://pusatprestasinasional.kemdikbud.go.id/event/sains-dan-teknologi/dikti/kontes-robot-indonesia-2023-2023-dikti>
- [3] KRI 2023. (2023). <https://kontesrobotindonesia.id/kri-2023.html>
- [4] Pengembangan Talenta, B., Pusat, I., Nasional, P., Pendidikan, K., & Teknologi, D. (n.d.). BUKU PEDOMAN KONTES ROBOT INDONESIA (KRI) TAHUN 2023.
- [5] Caysar, D. (2014). Pengaturan Pergerakan Robot Lengan *Smart Arm Robotic AX-12A* melalui Pendekatan *Geometry Based Kinematic* Menggunakan Arduino.