



**SISTEM KONTROL DAN *MONITORING* PADA PERUBAHAN POSISI
ROBOT KONTES ROBOT SEPAK BOLA INDONESIA (KRSBI) BERODA**

SKRIPSI

Oleh

Zaqi Armanovandi

NIM 161910201069

PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS JEMBER

2020



**SISTEM KONTROL DAN *MONITORING* PADA PERUBAHAN POSISI
ROBOT KONTES ROBOT SEPAK BOLA INDONESIA (KRSBI) BERODA**

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu
syarat untuk menyelesaikan Program Studi Teknik Elektro (S1)
dan mencapai gelar Sarjana Teknik

Oleh

Zaqi Armanovandi

NIM 161910201069

PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS JEMBER

2020

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Allah SWT, atas Rahmat dan Hidayahnya yang tiada habis-habisnya diberikan dalam mengerjakan skripsi ini.
2. Nabi Muhammad SAW yang telah membimbing kami dari zaman jahiliyah menuju zaman yang terang benderang yaitu addinul Islam.
3. Ayahanda Sugiarto, Ibunda Rahmawati, dan Ibunda Lilik Puspita yang senantiasa tiada henti-hentinya memberikan doa, dukungan, bimbingan, serta kasih sayang.
4. Dosen Pembimbing Utama, Dr. Azmi Saleh, S.T., M.T. serta Dosen Pembimbing Anggota, Suprihadi Prasetyono, S.T., M.T. atas kesabaran serta keikhlasan dalam membimbing penulis menyelesaikan skripsi ini.
5. Seluruh Dosen Teknik Elektro Universitas Jember yang telah memberikan bekal ilmu pengetahuan dan bimbingannya selama mengikuti pendidikan.
6. Keluarga Besar Himpunan Mahasiswa Elektro sebagai wadah penulis dalam memperoleh ilmu dan wawasan keorganisasian serta mendapat keluarga baru dalam mengemban tugas bersama.
7. Keluarga Besar UKM Robotika Teknik Universitas Jember sebagai wadah penulis memperoleh ilmu robotika dan wawasan keorganisasian serta mendapat keluarga baru dalam membuat robot dalam ajang lomba Kontes Robot Indonesia.
8. Seluruh Dulur Elektro 2016 yang telah memberikan dukungan serta doanya.
9. Sahabat-sahabatku tercinta yang telah memberikan semangat serta doanya.

MOTTO

“Allah akan meninggikan orang-orang yang beriman diantara kamu dan orang-orang yang berilmu beberapa derajat”

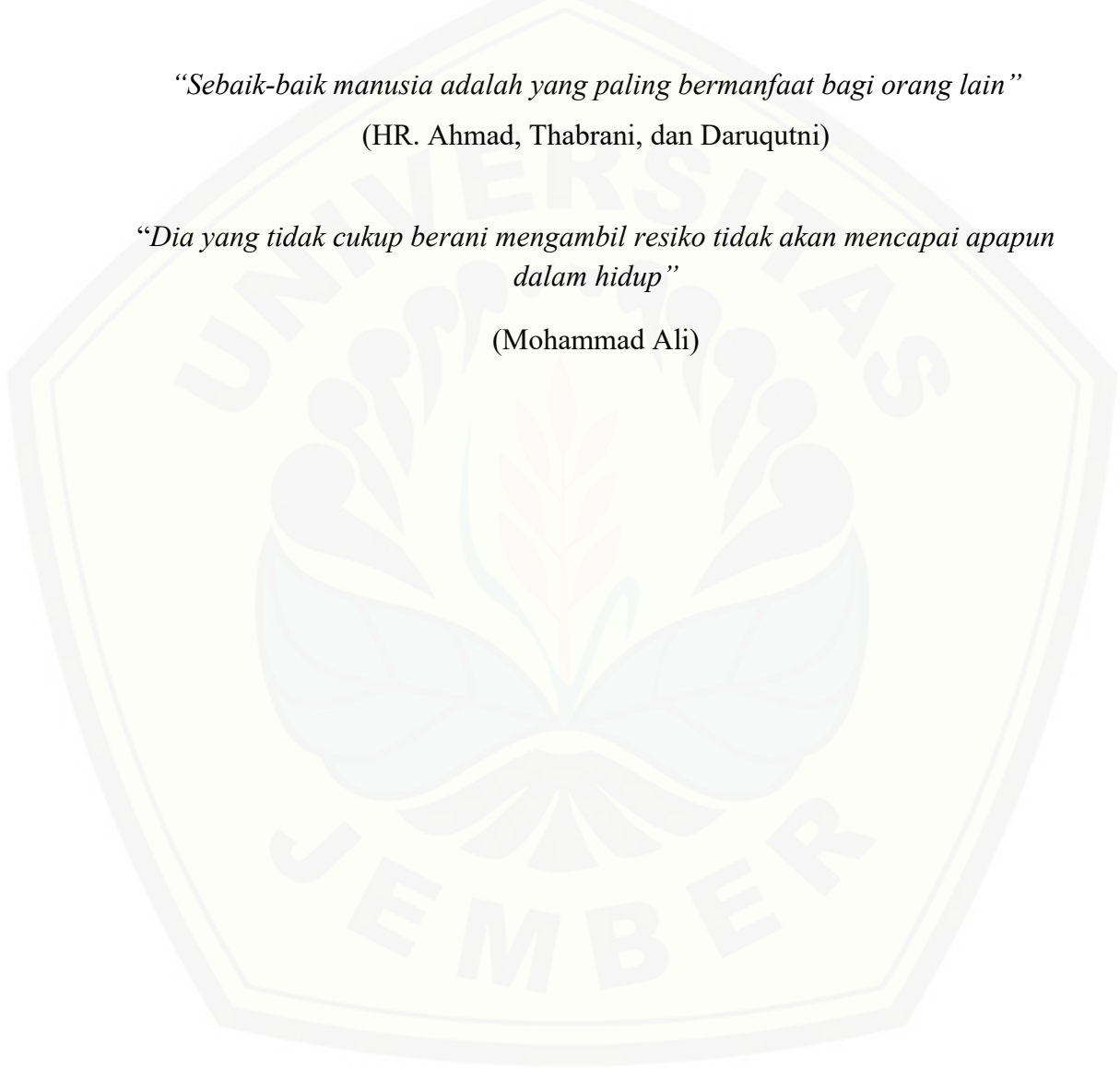
(QS. Al-Mujadalah: 11)

“Sebaik-baik manusia adalah yang paling bermanfaat bagi orang lain”

(HR. Ahmad, Thabrani, dan Daruqutni)

“Dia yang tidak cukup berani mengambil resiko tidak akan mencapai apapun dalam hidup”

(Mohammad Ali)



PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Zaqi Armanovandi

NIM : 161910201069

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Kontrol dan *Monitoring* pada Perubahan Posisi Robot Kontes Robot Sepak Bola Indonesia (KRSBI) Beroda” adalah benar-benar hasil karya sendiri, kecuali kutipan yang sudah saya sebutkan sumbernya, belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab penuh atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi. Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 8 Juli 2020

Yang menyatakan,

Zaqi Armanovandi

NIM 161910201069

SKRIPSI

**SISTEM KONTROL DAN *MONITORING* PADA PERUBAHAN POSISI
ROBOT KONTES ROBOT SEPAK BOLA INDONESIA (KRSBI) BERODA**

Zaqi Armanovandi

NIM 161910201069

Pembimbing

Dosen Pembimbing Utama : Dr. Azmi Saleh, S.T., M.T.

Dosen Pembimbing Anggota : Suprihadi Prasetyono, S.T., M.T.

PENGESAHAN

Skripsi berjudul “Sistem Kontrol dan *Monitoring* pada Perubahan Posisi Robot Kontes Robot Sepak Bola Indonesia (KRSBI) Beroda” karya Zaqi Armanovandi telah diuji dan disahkan pada:

Hari : Rabu

Tanggal : 8 Juli 2020

Tempat : Fakultas Teknik Universitas Jember

Tim Penguji

Pembimbing Utama,

Dr. Azmi Saleh, S.T., M.T.
NIP 197106141997021001

Penguji I,

Andi Setiawan, S.T., M.T.
NIP 196910101997021001

Pembimbing Anggota,

Suprihadi Prasetyono, S.T., M.T.
NIP 19700404199611001

Penguji II,

Wahyu Muldayani, S.T., M.T.
NRP 760016799

Mengesahkan

Dekan Fakultas Teknik
Universitas Jember,

Dr. Triwahju Hardiyanto, S.T., M.T.
NIP 197008261997021001

RINGKASAN

Sistem Kontrol dan *Monitoring* pada Perubahan Posisi Robot Kontes Robot Sepak Bola Indonesia (KRSBI) Beroda; Zaqi Armanovandi, 161910201069; 2020; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Dalam pertandingan KRSBI Beroda, robot harus dapat beroperasi secara otomatis dalam mengambil bola, menggiring bola, menendang bola, menghindari lawan, dan menuju ke posisi awal *start* sesuai dengan instruksi wasit, seperti *corner*, *golkick*, *penalty*, dan lainnya. Oleh karena itu, diperlukan sistem *monitoring* navigasi dan kontrol robot secara *wireless*, sehingga dapat memonitor perpindahan posisi robot secara *real time* dan memberi perintah kepada robot agar berpindah posisi sesuai yang diinginkan, juga dapat mengkoordinasi pergerakan antar robot secara otomatis.

Pada sistem kontrol dan *monitoring* terdiri dari satu MTU (*Master Terminal Unit*) dan dua RTU (*Remote Terminal Unit*) atau robot yang dikontrol. MTU mengontrol dan memonitor posisi kedua RTU atau robot. Komunikasi antara MTU dengan RTU menggunakan komunikasi TCP/IP secara *wireless* dan menggunakan bahasa sendiri yang dibuat oleh peneliti. Pada sistem kontrol MTU akan memberikan kode perintah pada setiap RTU, lalu kedua RTU akan menerjemahkan kode yang diberikan oleh MTU dan menjalankan kode perintah tersebut. Sedangkan pada sistem *monitoring* MTU akan mengirimkan kode *request* data ke tiap-tiap RTU, lalu tiap-tiap RTU akan membalas dengan mengirimkan koordinat (x,y) robot. MTU akan menerima kode tersebut dan menampilkannya koordinat (x,y) robot pada HMI (Human Machine Interface) dan menyimpannya pada *database* secara *real time*.

Hasil pengujian kontrol dan *monitoring* memiliki tingkat keberhasilan yang baik. Dimana robot bergerak sesuai dengan kontrol yang diberikan dengan waktu pengiriman kode perintah rata-rata sebesar 0,081 detik dan waktu respon motor rata-rata sebesar 0,229 detik saat tanpa penyimpanan data posisi robot dalam *database* serta dengan waktu pengiriman kode perintah rata-rata sebesar 0,375 detik

dan waktu respon motor rata-rata sebesar 0,52 detik saat dengan penyimpanan data posisi robot dalam *database*. Data koordinat (x,y) robot yang dikirim oleh RTU sesuai dengan data yang diterima oleh MTU dengan waktu pengiriman data rata-rata sebesar 0,085 detik saat tanpa penyimpanan data posisi robot pada *database* serta dengan waktu pengiriman data sebesar 0,377 detik saat dengan penyimpanan data posisi robot pada *database*



SUMMARY

Control and Monitoring System on Changes in Position of the Robot Kontes Robot Sepak Bola Indonesia (KRSBI) Beroda; Zaqi Armanovandi, 161910201069; 2020; Department of Electrical Engineering, Faculty of Engineering, University of Jember.

In a Wheeled KRSBI match, the robot must be able to operate automatically in picking up the ball, dribbling, kicking the ball, avoiding opponents, and heading to the starting position according to the referee's instructions, such as corners, goalkick, penalty, and others. Therefore, the robot's navigation and control monitoring system is needed wirelessly, so that it can monitor the movement of the robot's position in real time and give commands to the robot to move the position as desired, also can coordinate movement between robots automatically.

In the control and monitoring system consists of one MTU (Master Terminal Unit) and two RTU (Remote Terminal Unit) or controlled robot. MTU controls and monitors the position of both RTUs or robots. Communication between MTU and RTU uses TCP / IP communication wirelessly and uses its own language created by researchers. In the MTU control system will provide a command code on each RTU, then the two RTU will translate the code given by MTU and run the command code. Whereas the MTU monitoring system will send a request data code to each RTU, then each RTU will reply by sending the coordinates (x, y) of the robot. MTU will receive the code and display the coordinates (x, y) of the robot on the HMI (Human Machine Interface) and store it in the database in real time.

The results of control and monitoring testing have a good success rate. Where the robot moves according to the control given with an average command code sending time of 0.081 seconds and an average motor response time of 0.229 seconds when without robotic data storage in the database and with an average command code sending time of 0.375 seconds and average motor response time of 0.52 seconds when by storing robot position data in a database. Coordinate data (x, y) of robots sent by RTU in accordance with data received by MTU with an

average data transfer time of 0.085 seconds when without robotic position data storage in the database and with data sending time of 0.377 seconds when with position data storage robot on the database



PRAKATA

Segala puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi dengan Judul **“Sistem Kontrol dan Monitoring pada Perubahan Posisi Robot Kontes Robot Sepak Bola Indonesia (KRSBI) Beroda”**. Skripsi ini disusun guna memenuhi salah satu syarat menuntaskan pendidikan Strata Satu (S1) pada Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu penulis menyampaikan terima kasih kepada:

1. Allah SWT, atas kemurahan-Nya penulis dapat menyelesaikan skripsi ini.
2. Nabi Muhammad SAW, sebagai junjungan umat islam yang telah membimbing kami dari zaman jahiliyah menuju zaman yang terang benderang yaitu addinul Islam.
3. Ayahanda Sugiarto, Ibunda Rahmawati, dan Ibunda Lilik Puspita yang senantiasa tiada henti-hentinya memberikan doa, dukungan materi dan moral, bimbingan, serta kasih sayang.
4. Dr. Azmi Saleh, S.T., M.T. dan Suprihadi Prasetyono, S.T., M.T. selaku Dosen Pembimbing yang telah meluangkan waktu dan pikirannya dalam memberikan bimbingan, saran, dan motivasi demi terselesainya skripsi ini.
5. Andi Setiawan, S.T., M.T. dan Wahyu Muldayani, S.T., M.T. selaku Dosen Penguji yang telah memberikan masukan dan saran yang membangun demi penyempurnaan skripsi ini.
6. Ir.Khoirul Anam, S.T., M.T., Ph.D. selaku dosen pembimbing akademik yang telah membimbing dan menyemangati selama penulis menjadi mahasiswa.
7. Seluruh Dosen Teknik Elektro Universitas Jember yang telah memberikan bimbingan dan ilmu pengetahuan selama pendidikan di Universitas Jember.
8. Guru-guru yang telah membimbing serta memberikan ilmu pengetahuan dari taman kanak-kanak hingga sekolah menengah atas.
9. Keluarga Teknik Elektro 2016 “INDUKTRO” yang telah berproses bersama-sama selama menjadi mahasiswa Teknik Elektro Universitas Jember.

10. Keluarga besar Himpunan Mahasiswa Elektro dan UKM Robotika Teknik Universitas Jember. Yang telah bersama-sama menjalankan tugas dan saling mendukung satu sama lain juga bertukar ilmu dan pikiran dalam berproses selama menjalankan tugas..
11. Teman seperjuangan penghuni LPM, yaitu Cahya, Dandi, Erwin, Fikri, Syaiful, dan Tigo yang saling mendukung, saling membantu, saling menyemangati selama pengerjaan skripsi ini.
12. Serta seluruh pihak yang telah membantu penulisan dalam menyelesaikan skripsi yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih jauh dari kesempurnaan, karena sempurna hanya milik Allah SWT. Harapan penulis adalah semoga skripsi ini dapat bermanfaat bagi penulis dan pembaca.

Jember, 8 Juli 2020

Penulis

DAFTAR ISI

HALAMAN SAMPUL.....	ii
PERSEMBAHAN	iii
MOTTO	iv
PERNYATAAN.....	v
SKRIPSI	vi
PENGESAHAN.....	vii
RINGKASAN	viii
<i>SUMMARY</i>	x
PRAKATA.....	xii
DAFTAR ISI.....	xiv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xviii
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	4
BAB 2. TINJAUAN PUSTAKA	5
2.1 Sistem Kontrol	5
2.2 Sistem <i>Monitoring</i>	5
2.3 <i>Master Terminal Unit (MTU)</i>	5
2.2.1 <i>Human Machine Interface (HMI)</i>	5
2.2.2 <i>Database</i>	6
2.4 <i>Remote Terminal Unit (RTU)</i>	6
2.5 Teensy 4.0	6
2.6 Arduino Nano	8
2.7 <i>Odometry</i>	9
2.8 Sensor	10

2.7.1	Sensor Tegangan	10
2.7.2	<i>Rotary Encoder</i>	11
2.7.3	Sensor Arah CMPS12.....	11
2.7.4	Sensor Ultrasonik PING.....	12
2.9	Media Komunikasi.....	13
2.8.1	TCP/IP	13
2.8.2	Komunikasi Serial.....	14
2.8.3	Komunikasi I2C	15
3.1	Tempat Penelitian	16
3.2	Waktu Penelitian.....	16
3.3	Tahapan Penelitian	16
3.4	Alat dan Bahan Penelitian.....	19
3.5	Blok Diagram dan <i>Flowchart</i> Sistem Kontrol dan <i>Monitoring</i>	20
3.6	<i>Wiring Diagram</i> Sistem Kontrol dan <i>Monitoring</i>	27
3.7	Perancangan HMI.....	28
3.8	Perancangan Sistem Kontrol	30
3.9	Perancangan Sistem Monitoring	32
3.10	Perancangan Sistem Komunikasi.....	33
3.10.1	Media Komunikasi.....	33
3.10.1.1	Komunikasi TCP/IP.....	33
3.10.1.2	Komunikasi Serial.....	34
3.10.1.3	Komunikasi I2C	34
3.10.2	Komunikasi antara MTU dengan RTU	34
3.10.3	Komunikasi antara HMI RTU dengan Arduino Nano.....	35
3.10.4	Komunikasi antara Arduino Nano dengan Teensy 4.0.....	36
3.11	Sensor Tegangan	36
BAB 4. HASIL DAN PEMBAHASAN.....		39
4.1	Pengujian Sensor Tegangan.....	39
4.2	Pengujian Sensor Arah CMPS12.....	40
4.3	Pengujian Sensor Ultrasonik Ping.....	42
4.4	Pengujian Data Hasil <i>Odometry</i>	43

4.5	Pengujian Jarak Tendangan	45
4.6	Pengujian Sistem Komunikasi	46
4.6.1	Media Komunikasi	46
4.6.1.1	Komunikasi TCP/IP	46
4.6.1.2	Komunikasi Serial	46
4.6.1.3	Komunikasi I2C	47
4.6.2	Komunikasi antara MTU dengan RTU	47
4.6.3	Komunikasi antara HMI RTU dengan Arduino Nano	50
4.6.4	Komunikasi antara Arduino Nano dengan Teensy 4.0	54
4.7	Pengujian <i>Monitoring</i> Perubahan Posisi Robot	56
4.8	Pengujian Ketepatan Perintah Kontrol dari MTU ke RTU	59
4.9	Pengujian Kontrol Otomatis	61
BAB 5. KESIMPULAN DAN SARAN		63
5.1	Kesimpulan	63
5.2	Saran	64
DAFTAR PUSTAKA		65
LAMPIRAN		67

DAFTAR GAMBAR

Gambar 2. 1 Board Teensy 4.0	7
Gambar 2. 2 Board Arduino Nano	9
Gambar 2. 3 Skematik Sensor Tegangan	10
Gambar 2. 4 Sensor Rotary Encoder	11
Gambar 2. 5 Sensor CMPS12	12
Gambar 2. 6 Sensor Ultrasonik PING	12
Gambar 3. 1 Flowchart Penelitian	17
Gambar 3. 2 Blok Diagram Sistem Kontrol dan Monitoring	22
Gambar 3. 3 Flowchart Sistem Kontrol dan Monitoring	24
Gambar 3. 4 Three Omnidirectional Wheeled Robot	25
Gambar 3. 5 Wiring Diagram Perkabelan Sistem Kontrol dan Monitoring	27
Gambar 3. 6 Desain HMI pada MTU	28
Gambar 3. 7 Desain HMI pada RTU	29
Gambar 3. 8 (a) Blok Diagram Kontrol Posisi Robot, (b) Blok Diagram Kontrol Mengejar Bola, (c) Blok Diagram Kontrol Mencari Gawang Lawan, (d) Blok Diagram Kontrol Menendang Bola	30
Gambar 3. 9 Blok Diagram Sistem Monitoring	32
Gambar 3. 10 Rangkaian Pengujian Sensor Tegangan	38
Gambar 4. 1 Pengujian Sensor CMPS12	41
Gambar 4. 2 Pengujian Sensor Ultrasonik Ping	42
Gambar 4. 3 Pengujian Jarak Tendangan	45

DAFTAR TABEL

Tabel 2. 1 Spesifikasi Teensy 4.0	7
Tabel 2. 2 Spesifikasi Arduino Nano	8
Tabel 3. 1 Rencana Kegiatan Penelitian	16
Tabel 4. 1 Data Hasil Pengujian Sensor Pembagi Tegangan pada Vin Capacitor	40
Tabel 4. 2 Data Hasil Pengujian Sensor Arah CMPS12	41
Tabel 4. 3 Data Hasil Pengujian Sensor Ultrasonik Ping.....	43
Tabel 4. 4 Data Hasil Pengujian Jarak Tempuh Odometry.....	44
Tabel 4. 5 Data Hasil Pengujian Jarak Tendangan	45
Tabel 4. 6 Instruksi dan Data yang Dikirim oleh MTU ke RTU1	47
Tabel 4. 7 Instruksi dan Data yang Dikirim oleh MTU ke RTU2	48
Tabel 4. 8 Data yang Dikirim dari HMI RTU1 ke MTU	49
Tabel 4. 9 Data yang Dikirim dari HMI RTU2 ke MTU	50
Tabel 4. 10 Instruksi dan Data yang Dikirim dari HMI RTU1 ke Arduino Nano RTU1 .	51
Tabel 4. 11 Instruksi dan Data yang Dikirim dari HMI RTU2 ke Arduino Nano RTU2 .	52
Tabel 4. 12 Data yang Dikirim Arduino Nano2 RTU1 ke HMI RTU1	53
Tabel 4. 13 Data yang Dikirim Arduino Nano2 RTU2 ke HMI RTU2	54
Tabel 4. 14 Data yang Dikirim Arduino Nano1 RTU1 ke Teensy RTU1	55
Tabel 4. 15 Data yang Dikirim Arduino Nano1 RTU2 ke Teensy RTU2	55
Tabel 4. 16 Data yang Dikirim dari Teensy 4.0 RTU1 ke Arduino Nano2 RTU1	56
Tabel 4. 17 Data yang Dikirim dari Teensy 4.0 RTU2 ke Arduino Nano2 RTU2	56
Tabel 4. 18 Data Hasil Pengujian Transfer Data dari RTU ke MTU Tanpa Penyimpanan Data Posisi Robot pada Database.....	57
Tabel 4. 19 Data Hasil Pengujian Transfer Data dari RTU ke MTU dengan Penyimpanan Data Posisi Robot pada Database.....	58
Tabel 4. 20 Data Hasil Pengujian Ketepatan Perintah oleh MTU dan Respon RTU Tanpa Penyimpanan Data Posisi Robot pada Database.....	59
Tabel 4. 21 Data Hasil Pengujian Ketepatan Perintah oleh MTU dan Respon RTU dengan Penyimpanan Data Posisi Robot pada Database.....	60
Tabel 4. 22 Data Hasil Pengujian Keseluruhan	61

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Kontes Robot Indonesia (KRI) adalah ajang kompetisi rancang bangun dan rekayasa dalam bidang robotika yang diselenggarakan oleh Pusat Prestasi Nasional, Kementerian Pendidikan dan Kebudayaan Republik Indonesia. KRI terdiri dari enam divisi, yaitu divisi Kontes Robot Abu Indonesia (KRAI), divisi Kontes Robot Pemadam Api Indonesia (KRPAI), divisi Kontes Robot Seni Tari Indonesia (KRSTI), divisi Kontes Robot Sepak Bola Indonesia (KRSBI) Humanoid, divisi Kontes Robot Sepak Bola Indonesia (KRSBI) Beroda, dan divisi baru Kontes Robot Tematik Indonesia (KRTMI). (Kemendikbud, 2020).

Divisi KRSBI Beroda 2020 bertema “Sepak Bola Robot Menuju Liga Sepak Bola Robot Tahun 2050” yang diselenggarakan berdasarkan aturan yang dilakukan di *RoboCup Middle Size League* (MSL), dengan menyesuaikan kondisi di Indonesia, misalnya pada ukuran lapangan dan lainnya. (Kemendikbud, 2020). Pada divisi ini terdiri dari tiga robot otomatis (*autonomous wheeled robot*), satu sebagai kiper dan yang lain sebagai penyerang. Selama pertandingan, robot harus dapat beroperasi secara otomatis dalam mengambil bola, menggiring bola, menendang bola, menghindari lawan, dan menuju ke posisi awal *start* sesuai dengan instruksi wasit, seperti *corner*, *golkick*, *penalty*, dan lainnya.

Berdasarkan peraturan pertandingan yang telah ditetapkan seperti di atas, diperlukan sistem *monitoring* navigasi dan kontrol robot secara *wireless*, sehingga dapat memonitor perpindahan posisi robot secara *real time* dan memberi perintah kepada robot agar berpindah posisi sesuai yang diinginkan, juga dapat mengkoordinasi pergerakan antar robot secara otomatis.

Berdasarkan keadaan di atas maka penulis melakukan penelitian tentang sistem kontrol dan *monitoring* pada perpindahan posisi robot KRSBI Beroda. Sistem ini dibangun menggunakan microsoft visual studio 2010 sebagai HMI (*Human Machine Interface*), Microsoft Access 2016 sebagai *database* dan Teensy 4.0 sebagai pengolah data posisi robot dengan menggunakan metode odometry dan kontrol aktuator pada robot.

Pada penelitian terkait sebelumnya dilakukan oleh Aulia Aditya Rachman pada tahun 2017 dengan judul “Sistem Perencanaan Rute Gerak Pada Robot Sepak Bola Beroda”.dimana hanya meneliti tentang perencanaan rute gerak robot sepak bola beroda dari titik awal robot bergerak sampai titik dimana bola berada. Kemudian penelitian yang terkait juga dilakukan oleh Ihsan Syarifuddin dkk pada tahun 2017 “Pergerakan Otomatis Robot Sepak Bola Beroda Melalui Komunikasi dengan Referee Box Menggunakan Base Station” pada penelitiannya hanya menguji pengiriman data dari *RefBox* ke Robot tanpa pergerakan robot maupun koordinasi antar robot. Selanjutnya adalah penelitian oleh S.Sankhar ReddyCH. dkk pada tahun 2020 dengan judul “Quanta – A Platform for Rapid Control and Monitoring of Heterogeneous Robots” dimana memiliki kecepatan kontrol dan *monitoring* sebesar 471,3127 milidetik dengan menggunakan modul radio dan pada penelitiannya hanya mengirimkan data saja tidak menjelaskan tentang pergerakan robot maupun koordinasi antar robot.

1.2 Rumusan Masalah

Dalam penelitian ini ada beberapa hal yang menjadi rumusan masalah untuk menjadi rujukan apa yang akan dilakukan dan diteliti, diantaranya:

1. Bagaimana cara *monitoring* perubahan posisi robot secara *real time*?
2. Bagaimana cara mengontrol perubahan posisi robot?
3. Bagaimana cara menghubungkan Teensy, Arduino dan PC dengan sistem kontrol dan *monitoring* perubahan posisi robot?

1.3 Batasan Masalah

Berdasarkan uraian rumusan masalah tersebut, agar penelitian ini lebih terfokuskan maka pembahasan pada penelitian ini perlu dibatasi. Batasan masalah tersebut antara lain:

1. Sistem hanya melakukan *monitoring* dan pengontrolan perubahan posisi robot antara *Master Terminal Unit* (MTU) dengan *Remote Terminal Unit* (RTU).

2. Media komunikasi menggunakan TCP/IP secara *wireless* dengan router WIFI antara *Remote Terminal Unit* (RTU) dengan *Master Terminal Unit* (MTU) dengan jarak maksimal 9,5 meter sesuai dengan ukuran lapangan pertandingan KRSBI yaitu 9 kali 6 meter dan peraturan KRSBI dimana kontrol robot harus berada di belakang gawang sendiri.
3. Metode *odometry* digunakan untuk mengolah data perubahan posisi robot dengan sensor *rotary encoder* yang akan dikirim dari *Remote Terminal Unit* (RTU) ke *Master Terminal Unit* (MTU).
4. Kontrol PID digunakan pada kontrol pergerakan robot ke posisi yang dituju dengan *feedback* dari sensor CMPS12 dan dua *rotary encoder* yang diolah menggunakan metode *odometry*, kontrol robot mengejar bola dengan *feedback* dari kamera, dan kontrol robot mencari gawang lawan dengan *feedback* dari sensor CMPS12.
5. Robot menendang bola menggunakan solenoid dimana memanfaatkan medan elektromagnetik menjadi energi gerakan. Solenoid tersebut menggunakan daya dari *capacitor bank*. *Capacitor bank* sendiri di *supply* oleh battery lippo 16 V yang dinaikkan dulu tegangannya menjadi 400 V menggunakan *boost converter*. kapasitor yang digunakan untuk *capacitor bank* adalah kapasitor 4800Uf 400V untuk tendangan keras dan kapasitor 1000Uf untuk tendangan pelan.
6. Robot membaca bola dengan menggunakan dua buah kamera dan diolah menggunakan pengolahan citra menghasilkan data jarak dan sudut bola terhadap robot.

1.4 Tujuan Penelitian

Penelitian ini memiliki beberapa tujuan yang ingin dicapai sebagai pencapaian akhir, diantaranya:

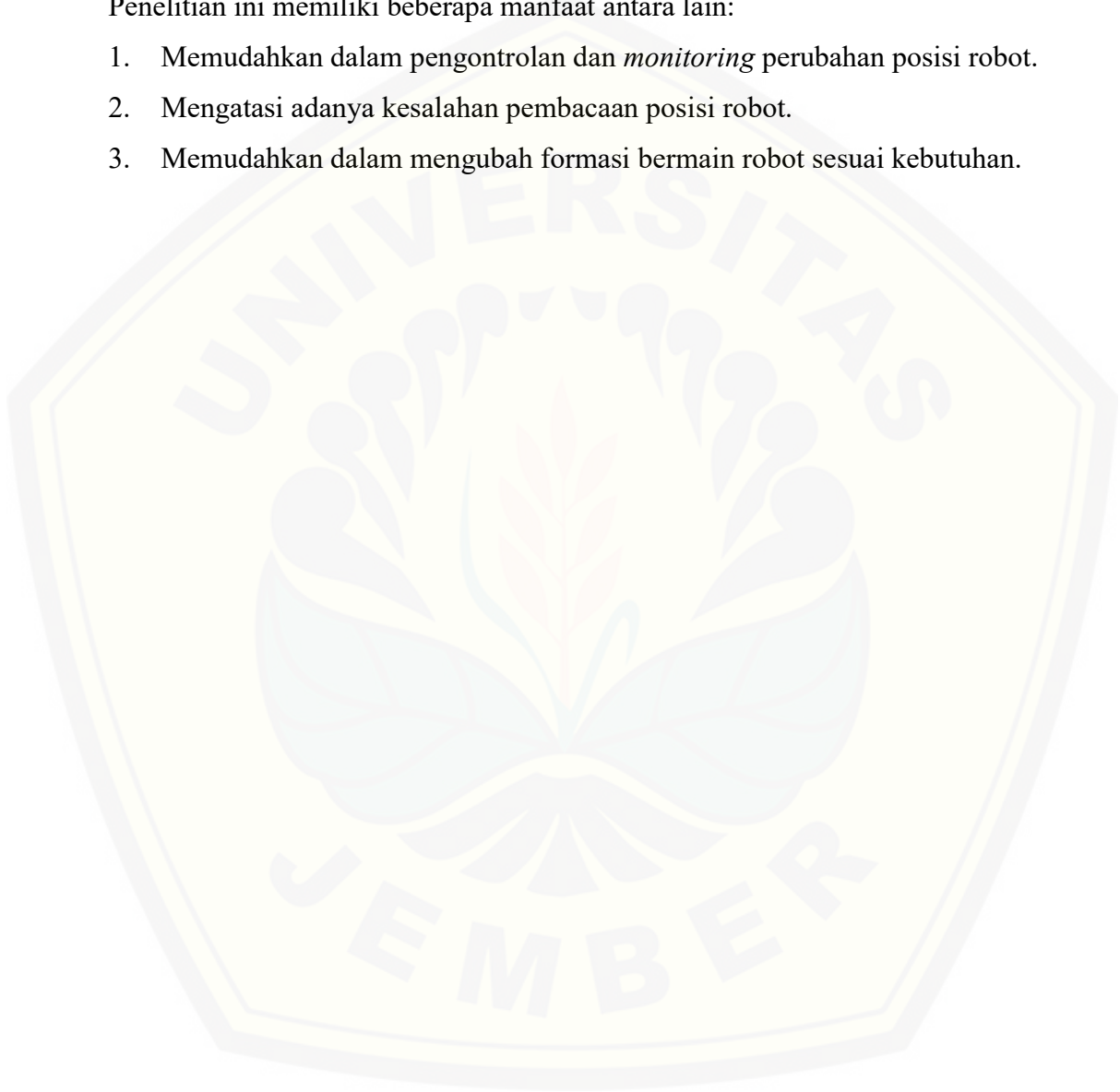
1. Membuat sistem *monitoring* pada perubahan posisi robot secara *real time*.
2. Dapat melakukan pengontrolan pada perubahan posisi robot.
3. Membuat sistem kontrol dan *monitoring* perubahan posisi robot dengan Teensy dan Arduino, dimana Teensy berfungsi sebagai sistem kontrol dan pengolah

data sensor pada robot dan Arduino sebagai penghubung antara HMI dengan Teensy.

1.5 Manfaat Penelitian

Penelitian ini memiliki beberapa manfaat antara lain:

1. Memudahkan dalam pengontrolan dan *monitoring* perubahan posisi robot.
2. Mengatasi adanya kesalahan pembacaan posisi robot.
3. Memudahkan dalam mengubah formasi bermain robot sesuai kebutuhan.



BAB 2. TINJAUAN PUSTAKA

2.1 Sistem Kontrol

Sistem kontrol atau sistem kendali adalah kumpulan alat-alat yang tergabung menjadi sebuah sistem untuk mengendalikan, memerintah dan mengatur keadaan suatu sistem. Dalam robot sistem kontrol terdiri dari mikrokontroler sebagai otak, aktuator penggerak robot, dan sensor-sensor.

2.2 Sistem Monitoring

Sistem *monitoring* adalah sebuah sistem yang berfungsi sebagai proses pengumpulan data dan melakukan analisis terhadap data-data tersebut. Sistem monitoring tidak lepas dari sistem komunikasi untuk mengirimkan data dari jarak yang jauh. Sistem komunikasi dapat menggunakan *wire system* maupun *wireless system*.

2.3 Master Terminal Unit (MTU)

Master Terminal Unit atau MTU adalah sebuah sistem komputer, bisa berupa komputer, *Programmable Logic Controller* (PLC), atau *Microcontroller* yang berfungsi untuk mengolah data dari *Remote Terminal Unit* (RTU) lalu diterjemahkan dan diberikan ke *Human Machine Interface* (HMI). Di sisi lain MTU bertugas untuk mengambil data dari tiap-tiap RTU (*Remote Terminal Unit*), jika RTU lebih dari satu. Sistem pengambilan data dari tiap-tiap RTU disebut *polling*. MTU dan HMI dapat dijadikan 1 bagian ketika MTU menggunakan komputer yang sama dengan HMI.

2.2.1 Human Machine Interface (HMI)

Human Machine Interface atau HMI adalah subsistem dari SCADA yang berfungsi untuk menampilkan data hasil pengukuran RTU ataupun proses yang sedang terjadi pada keseluruhan sistem di RTU. HMI merupakan sebuah *software* pada komputer berbasis grafis yang berfungsi untuk mempermudah pengawasan

(*supervisory*) kepada operator. HMI mengubah data-data dan angka ke dalam animasi, grafik, dan bentuk yang mudah diterjemahkan oleh operator.

2.2.2 Database

Database adalah kumpulan data atau informasi yang disimpan pada komputer secara sistematis sehingga dapat melihat, mengubah, atau menambahkan data atau informasi dengan menggunakan suatu program komputer.

2.4 Remote Terminal Unit (RTU)

Remote Terminal Unit atau RTU subsistem SCADA yang berfungsi sebagai terminal-terminal (semacam stasiun data) dari hasil pengukuran, pengendalian, pemantauan status, dan lain-lain. RTU juga berfungsi untuk menterjemahkan, mengkonversi, menghitung sinyal dari transducer seperti pengukuran arus listrik, *flow static pressure*, *differensial pressure*, *temperature*, dan lain-lain. Dari hasil pengukuran tersebut, hal yang dilakukan RTU adalah melakukan kendali jika merupakan sistem kendali kemudian mentransfer data ke MTU atau langsung mentransfer data ke MTU jika sistem RTU bukan merupakan sistem kendali. RTU juga dapat berfungsi sebagai pengatur set point yang dikirimkan dari MTU ke RTU tersebut.

2.5 Teensy 4.0

Teensy adalah modul mikrokontroler 32bit ARM yang memiliki fleksibilitas mumpuni, kaya akan fitur dengan dimensi yang kecil serta *breadboard-friendly* yang dikembangkan oleh perusahaan PJRC dengan perancangannya yaitu Paul Stoffregen yang merupakan *co-founder* perusahaan tersebut. Teensy terinspirasi dari mikrokontroler Arduino yang penggunaannya mudah serta memiliki fitur yang lengkap.

Teensy 4.0 merupakan produk terbaru teensy dimana tertanam prosesor ARM Cortex-M7 berjalan pada kecepatan 600 MHz dengan chip NXP iMXRT1062. Sehingga Teensy 4.0 dapat menjalankan program lebih cepat dari

pendahulunya yaitu lima kali lebih cepat dari Teensy 3.6 dan tujuh kali lebih cepat dari Teensy 3.5.

Berikut adalah tabel spesifikasi dari Teensy 4.0 yang dapat dilihat pada Tabel 2.1 di bawah ini.

Tabel 2. 1 Spesifikasi Teensy 4.0

Prosesor	ARM Cortex-M7
Tegangan Operasi	3,3 V
Tegangan Input	3,6-5,5 V
USB Port	2 buah, dengan kecepatan 480 Mbps
I/O Digital	40 buah (mampu digunakan sebagai pin interrupt)
I/O PWM Digital	31 buah
Analog Input	14 buah
Komunikasi	3 buah CAN Bus, 2 buah I2S Digital Audio, 3 buah SPI, 3 buah I2C, dan 7 buah Serial
RTC Date/Time	1 buah
Power On/Off Management	ada
RAM	1024 Kb
Flash Memory	2048 Kb
Kecepatan Clock	600 Mhz

(Sumber: www.pjrc.com)

Berikut merupakan gambar dari *board* teensy 4.0 yang dapat dilihat pada Gambar 2.1 di bawah ini.



Gambar 2. 1 Board Teensy 4.0

(Sumber: www.pjrc.com)

2.6 Arduino Nano

Arduino Nano adalah modul mikrokontroler arduino terkecil yang didesain dan diproduksi oleh Gravitech. Arduino Nano menggunakan mikrokontroler ATmega 328. Modul ini memiliki dua puluh dua pin *input-output* (22 I/O) di mana enam diantaranya dapat digunakan sebagai *output pulse width modulation* (PWM), delapan *input* analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, dan tombol *reset*. Arduino nano bekerja pada tegangan 5VDC dan dapat menerima arus 40mA.

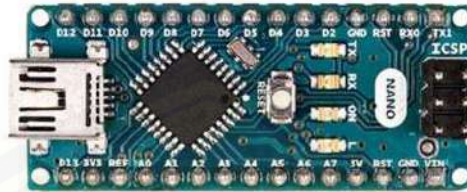
Berikut adalah tabel spesifikasi dari Arduino Nano yang dapat dilihat pada Tabel 2.2 di bawah ini.

Tabel 2. 2 Spesifikasi Arduino Nano

Mikrokontroler	Atmega 328
Tegangan Operasi	5 V
Tegangan Input	7-12 V
Tegangan Input (Batas)	6-20 V
I/O Digital	22 (dengan tersedia 6 buah PWM output)
PWM digital I/O	6 buah
Input Analog	8 buah
Arus DC per I/O pin	40 Ma
Arus DC untuk pin 3.3	50 Ma
Flash Memory	32 KB (ATmega328P) Dengan 0.5 KB digunakan oleh Bootloader
SRAM	2 Kb (ATmega328p)
EEPROM	1 Kb (ATmega328p)
Kecepatan Clock	16 MHz

(Sumber: www.arduino.cc)

Berikut merupakan gambar dari *board* teensy 4.0 yang dapat dilihat pada Gambar 2.1 di bawah ini.



Gambar 2. 2 Board Arduino Nano
(Sumber: www.arduino.cc)

2.7 Odometry

Odometry adalah metode kontrol yang menggunakan data dari sensor posisi untuk memperkirakan perubahan posisi dari waktu ke waktu. *Odometry* digunakan untuk memperkirakan posisi robot terhadap posisi awal robot bergerak pada koordinat kartesian (Basori, 2014).

Odometry didasarkan pada asumsi bahwa revolusi roda dapat diterjemahkan ke dalam perpindahan linier relatif ke lantai. Asumsi ini hanya berlaku terbatas. Salah satu contohnya adalah roda selip, jika satu roda itu terpeleset maka *encoder* pada roda itu akan menghilangkan revolusi roda meskipun revolusi ini tidak akan sesuai dengan perpindahan linier roda. Selain kasus total selip ini, ada beberapa lainnya, alasan yang lebih halus untuk ketidakakuratan dalam terjemahan bacaan roda *encoder* menjadi gerak linier. Semua sumber kesalahan masuk ke dalam salah satu dua kategori yakni kesalahan sistematis dan kesalahan *nonsystematic* (Borenstein, 1996).

Kesalahan sistematis meliputi diameter roda yang tidak sama, roda tidak berbentuk lingkaran sempurna, ketidakpastian titik gesekan pada roda, resolusi *encoder* yang terbatas, dan juga *sampling rate* yang terbatas. Kesalahan *nonsystematic* meliputi lantai yang tidak rata, roda selip dan juga terganjal benda yang tidak terduga (Rokhim, 2017).

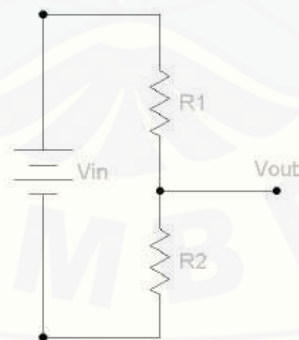
2.8 Sensor

Sensor adalah *hard device* yang digunakan untuk merubah suatu nilai tertentu menjadi satuan sinyal sehingga dapat diterjemahkan oleh rangkaian elektronik tertentu. Sensor sendiri merupakan komponen utama dari suatu transduser, dimana transduser merupakan sistem yang melengkapi agar sensor tersebut mempunyai keluaran sesuai yang kita inginkan dan dapat langsung diketahui nilainya (Rokhim, 2017).

Sensor termasuk transduser yang digunakan untuk mengubah besaran mekanik, kimia, suhu, cahaya, dan magnetis menjadi tegangan dan arus listrik. Sensor sering dipergunakan untuk pendeteksian saat melakukan pengukuran atau kontrol suatu sistem.

2.7.1 Sensor Tegangan

Sensor tegangan adalah suatu rangkaian pembagi tegangan (*voltage divider*) yang berfungsi untuk membaca nilai tegangan dengan mengubah tegangan yang lebih besar (tegangan yang akan diukur) menjadi tegangan yang lebih kecil agar dapat dimasukkan dan dibaca di pin mikrokontroler yang sudah memiliki ADC (*Analog to Digital Converter*).



Gambar 2. 3 Skematik Sensor Tegangan

Keterangan:

V_{in} : Tegangan yang akan diukur (tegangan input)

V_{out} : Tegangan yang masuk di pin analog mikrokontroler (tegangan output)

R : Resistansi

2.7.2 Rotary Encoder

Rotary encoder adalah komponen elektronika yang bekerja secara mekanik dan menggunakan sensor optik maupun kontak yang menghasilkan pulsa-pulsa yang dapat diubah kedalam suatu nilai posisi atau arah. Pada Gambar 2.10 menunjukkan prinsip kerja secara umum dari *rotary encoder*, dimana sebuah *disk* tipis dan LED yang ditempatkan sedemikian rupa sehingga cahayanya tetap terfokus pada *disk* tersebut. Sebuah *phototransistor* ditempatkan pada sisi lain dari *disk* sehingga dapat mendeteksi cahaya dari LED. *disk* tersebut diposisikan pada *shaft* yang bergerak. Dimana pergerakan *disk* tersebut sesuai dengan pergerakan *shaft*, sehingga ketika *shaft* berputar, maka *disk* ikut berputar. Ketika *disk* yang di atasnya ditempatkan LED yang memancarkan cahaya yang terfokus terhadap *phototransistor*, *phototransistor* akan berada dalam keadaan saturasi yang keluarannya berupa pulsa gelombang kotak (Banodin, 2006).

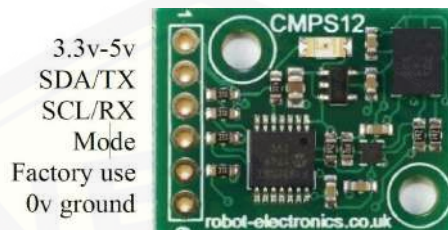


Gambar 2. 4 Sensor *Rotary Encoder*
(Sumber: www.encoders-uk.com)

2.7.3 Sensor Arah CMPS12

CMPS12 adalah sensor kompas magnetik dan IMU (*Inertia Measurement Unit*) yang didesain khusus dalam bidang robotik untuk tujuan navigasi robot. Kompas ini ditenagai chip sensor 9-axis IMU BNO055 buatan Bosch yang terdiri dari *3-axis magnetometer*, *3-axis gyroscop*, dan *3-axis accelerometer*. Terdapat Kalman filter yang mengkombinasikan sensor *gyroscop* dan *accelerometer* untuk menghilangkan kesalahan pembacaan akibat dari kemiringan PCB (*Printed Circuit Board*) yang diproses oleh prosesor 16 bit yang terdapat pada CMPS12. Prosesor tersebut melakukan perhitungan kompleks sehingga dihasilkan *output* 0 hingga 359,9 derajat,

Modul CMPS12 ini membutuhkan suplai tegangan sebesar 3,3VDC hingga 5VDC dengan konsumsi arus sekitar 18mA. Ada dua cara untuk membaca . Yaitu melalui protokol serial atau melalui protokol I2C .



Gambar 2. 5 Sensor CMPS12
(Sumber: www.robot-electronics.co.uk)

2.7.4 Sensor Ultrasonik PING

Sensor ini merupakan sensor ultrasonik yang mengubah besaran bunyi menjadi besaran listrik dan dari besaran listrik ke besaran bunyi. Sensor ini berfungsi sebagai pendeteksi jarak dari suatu objek dengan cara memancarkan gelombang ultrasonik dengan frekuensi 40 KHz dan kemudian mendeteksi pantulannya. Sensor ini hanya dapat mengukur jarak antara 3 cm sampai 300 cm. *Output* dari sensor ini berupa pulsa yang lebarnya mempresentasikan jarak. Dimana lebar pulsanya bervariasi dari 115 us sampai 18,6 ms.



Gambar 2. 6 Sensor Ultrasonik PING
(Sumber: www.parallax.com)

2.9 Media Komunikasi

SCADA tidak pernah lepas dengan komunikasi. Komunikasi yang terdapat pada SCADA yaitu menghubungkan RTU dengan MTU (Winarno, 2010).

Secara sederhana, sebuah media komunikasi adalah sebuah perantara dalam menyampaikan sebuah informasi dari komunikator kepada komunikan yang bertujuan agar efisien dalam menyebarkan informasi atau pesan. Komunikasi merupakan bentuk percakapan yang berlangsung atas dasar persamaan persepsi. Komunikasi dalam bahasa Inggris *communication* berasal dari kata Latin *communication* dan berasal dari kata *communis* yang berarti sama.

2.8.1 TCP/IP

TCP/IP (*Transmission Control Protocol / Internet Protocol*) merupakan gabungan protokol TCP (*Transmission Control Protocol*) dan IP (*Internet Protocol*) sebagai kesatuan kelompok protokol untuk mengatur proses tukar-menukar data atau komunikasi data dari satu komputer ke komputer lain di dalam jaringan internet dan memastikan data terkirim ke alamat yang dituju. Protokol ini merupakan kumpulan beberapa protokol (*protocol suite*) karena protokol ini tidak dapat berdiri sendiri. Protokol ini mampu bekerja dan diterapkan pada lintas perangkat lunak pada berbagai sistem operasi sehingga protokol ini yang paling banyak digunakan saat ini. Terdapat dua buah skema pengalamatan yang dapat digunakan untuk mengidentifikasi sebuah komputer dalam sebuah jaringan atau jaringan dalam sebuah Internet (*Interconnection Network*) pada protokol TCP/IP, yaitu pengalamatan IP dan FQDN (*Fully Qualified Domain Name*).

Pengalamatan IP berupa alamat logis yang terdiri atas 32-bit (empat oktet berukuran 8-bit) yang umumnya ditulis dalam format *www.xxx.yyy.zzz*. Dengan menggunakan *subnet mask* yang diasosiasikan dengannya, sebuah alamat IP pun dapat dibagi menjadi dua bagian, yakni *Network Identifier* (NetID) yang dapat mengidentifikasi jaringan lokal dalam sebuah *internetwork* dan *Host identifier* (HostID) yang dapat mengidentifikasi *host* dalam jaringan tersebut. Sebagai contoh, alamat 205.116.008.044 dapat dibagi dengan menggunakan subnet mask 255.255.255.000 ke dalam *Network*

ID 205.116.008.000 dan *Host* DI 44. Alamat IP merupakan kewajiban yang harus ditetapkan untuk sebuah *host*, yang dapat dilakukan secara manual (statis) atau menggunakan *Dynamic Host Configuration Protocol* (DHCP) (dinamis).

FQDN (*Fully Qualified Domain Name*) merupakan alamat yang direpresentasikan dalam nama alfanumerik yang diekspresikan dalam bentuk `<nama_host>.<nama_domain>`, di mana `<nama_domain>` mengidentifikasi jaringan di mana sebuah komputer berada, dan `<nama_host>` mengidentifikasi sebuah komputer dalam jaringan. Pengalamatan FQDN digunakan oleh skema penamaan domain Domain Name System (DNS). Sebagai contoh, alamat FQDN `id.wikipedia.org` merepresentasikan sebuah host dengan nama "id" yang terdapat di dalam domain jaringan "wikipedia.org". Nama domain `id.wikipedia.org` merupakan *second-level domain* yang terdaftar di dalam *top-level domain* ".org", yang terdaftar dalam root DNS, yang memiliki nama "." (titik). Penggunaan FQDN lebih bersahabat dan lebih mudah diingat ketimbang dengan menggunakan alamat IP. Akan tetapi, dalam TCP/IP, agar komunikasi dapat berjalan, FQDN harus diterjemahkan terlebih dahulu (proses penerjemahan ini disebut sebagai resolusi nama) ke dalam alamat IP dengan menggunakan *server* yang menjalankan DNS, yang disebut dengan *Name Server* atau dengan menggunakan berkas *hosts* (`/etc/hosts` atau `%systemroot%\system32\drivers\etc\hosts`) yang disimpan di dalam mesin yang bersangkutan.

2.8.2 Komunikasi Serial

Komunikasi serial adalah komunikasi yang pengiriman datanya per-bit secara berurutan dan bergantian. Komunikasi ini mempunyai suatu kelebihan yaitu hanya membutuhkan satu jalur dan kabel yang sedikit dibandingkan dengan komunikasi paralel. Pada prinsipnya komunikasi serial merupakan komunikasi dimana pengiriman data dilakukan per bit sehingga lebih lambat dibandingkan komunikasi paralel, atau dengan kata lain komunikasi serial merupakan salah satu metode komunikasi data di mana hanya satu bit data yang dikirimkan melalui seuntai kabel pada suatu waktu tertentu. Pada dasarnya komunikasi serial adalah

kasus khusus komunikasi paralel dengan nilai $n = 1$, atau dengan kata lain adalah suatu bentuk komunikasi paralel dengan jumlah kabel hanya satu dan hanya mengirimkan satu bit data secara simultan. Hal ini dapat disandingkan dengan komunikasi paralel yang sesungguhnya di mana n -bit data dikirimkan bersamaan, dengan nilai umumnya $8 \leq n \leq 128$.

Komunikasi serial ada 2 macam, *asynchronous serial* dan *synchronous serial*. *Synchronous serial* adalah komunikasi dimana hanya ada satu pihak (pengirim atau penerima) yang menghasilkan clock dan mengirimkan clock tersebut bersama-sama dengan data. Contoh penggunaan *synchronous serial* terdapat pada transmisi data keyboard. *Asynchronous serial* adalah komunikasi dimana kedua pihak (pengirim dan penerima) masing-masing menghasilkan clock namun hanya data yang ditransmisikan, tanpa clock. Agar data yang dikirim sama dengan data yang diterima, maka kedua frekuensi clock harus sama dan harus terdapat sinkronisasi. Setelah adanya sinkronisasi, pengirim akan mengirimkan datanya sesuai dengan frekuensi clock pengirim dan penerima akan membaca data sesuai dengan frekuensi clock penerima. Contoh penggunaan asynchronous serial adalah pada Universal Asynchronous Receiver Transmitter (UART) yang digunakan pada serial port (COM) komputer.

2.8.3 Komunikasi I2C

I2C (*Inter Integrated Circuit*) adalah komunikasi standar serial menggunakan dua saluran dengan desain khusus untuk mengirim maupun menerima data. I2C terdiri dari saluran SDA (*Serial Data*) yang membawa data-data antara pengirim (*transmitter*) dan penerima (*receiver*) dan SCL (*Serial Clock*). Pada sistem I2C piranti yang terhubung dalam akses bus serial dapat dioperasikan sebagai *Master* atau *Slave*. *Master* merupakan piranti yang mengendalikan operasi pengiriman data, sedangkan *Slave* merupakan piranti yang dikendalikan oleh *Master* dalam operasi pengiriman data.

BAB 3. METODOLOGI PENELITIAN

3.1 Tempat Penelitian

Penelitian ini dilakukan di Laboratorium Elektronka Terapan, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember, Jl Slamet Riyadi No. 62, kecamatan Patrang, kabupaten Jember. Penelitian juga dilakukan di LPM Bhayangkara Universitas Jember, Jl. Veteran No. 3, kecamatan Patrang, Kabupaten Jember.

3.2 Waktu Penelitian

Penelitian ini direncanakan akan dimulai pada bulan Desember 2019 sampai selesai dengan rincian sebagai berikut:

Tabel 3. 1 Rencana Kegiatan Penelitian

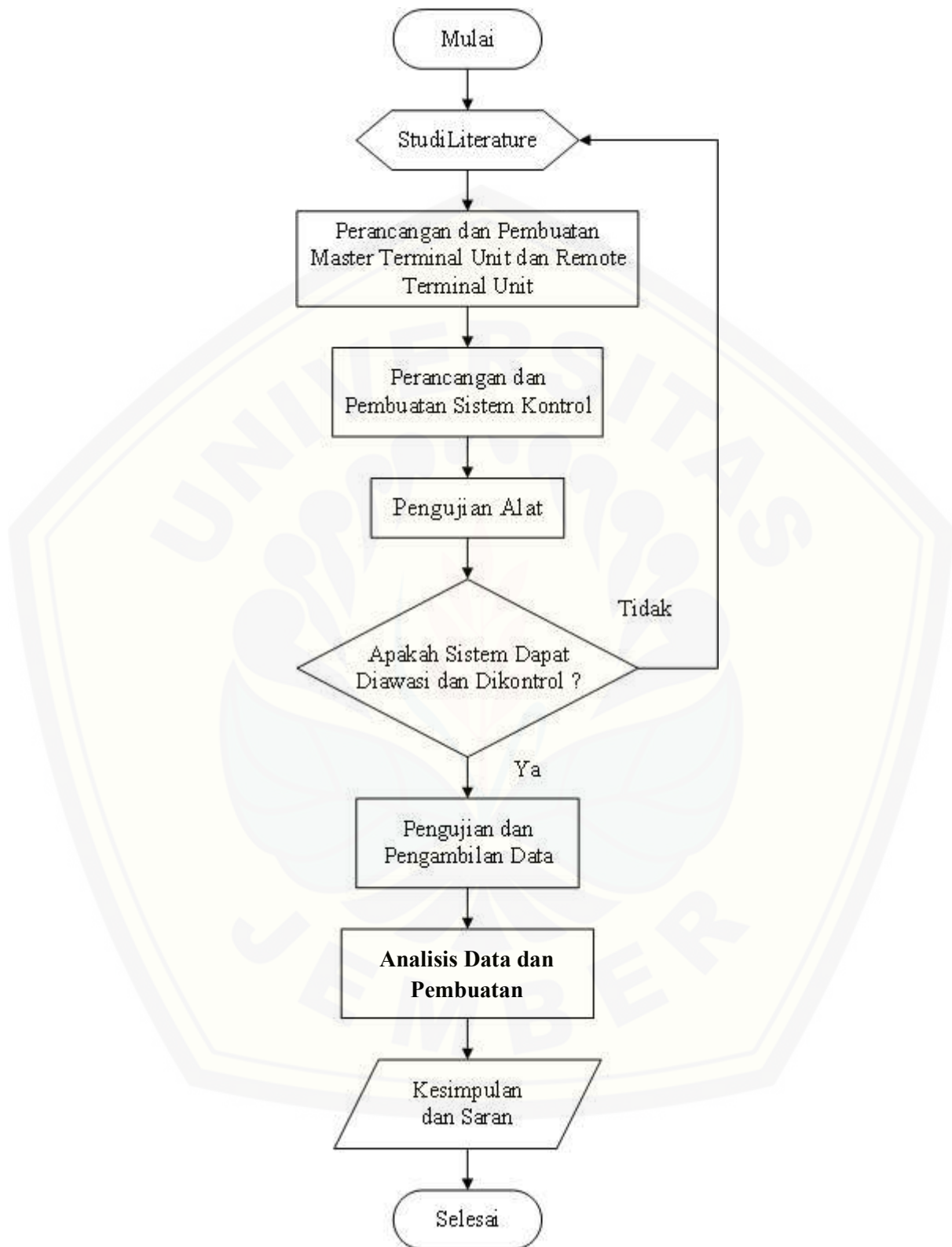
No	Kegiatan	Bulan Ke-/Minggu															
		Bulan 1				Bulan 2				Bulan 3				Bulan 4			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Studi Literatur	■	■	■	■												
2.	Penelitian					■	■	■	■	■	■	■	■				
3.	Analisis Data									■	■	■	■				
4.	Pembahasan													■	■	■	■
5.	Pembuatan Laporan													■	■	■	■

Keterangan:

■ : Kegiatan dilaksanakan

3.3 Tahapan Penelitian

Berikut merupakan tahap-tahap dalam penelitian keseluruhan sebagaimana dapat dinyatakan dalam diagram alir yang terdapat pada Gambar 3.1.

Gambar 3. 1 *Flowchart* Penelitian

Tahapan-tahapan yang dilakukan pada penelitian “Sistem Kontrol dan *Monitorig* pada Perubahan Posisi Robot Kontes Robot Sepak Bola Indonesia (KRSBI) Beroda “ adalah sebagai berikut :

1. Studi Literatur

Tahap pertama yaitu dengan mencari sumber literatur yang berkaitan dengan penelitian ini. Didapatkan teori antara lain yaitu *odometry* dari penelitian yang diteliti oleh S. Basori dkk pada tahun 2014 yang berjudul “Implementasi *Odometry* pada Robot Otomatis Kontes Robot Abu Indonesia” selanjutnya yaitu pengiriman data dari penelitian yang dilakukan oleh T. Ardhiansyah dkk pada tahun 2017 yang berjudul “Pergerakan Otomatis Robot Sepak Bola Beroda Melalui Komunikasi dengan Referee Box Menggunakan Base Station” dan lainnya yang terdapat pada DAFTAR PUSTAKA. Dengan adanya sumber literatur tersebut, diharapkan perancangan dan pelaksanaan dapat sesuai dengan yang diharapkan.

2. Perancangan dan Pembuatan *Master Terminal Unit* dan *Remote Terminal Unit*

Tahap kedua yaitu merancang MTU yang digunakan sebagai pusat kendali dan *database*. Sedangkan RTU digunakan untuk mengontrol perubahan posisi robot.

3. Perancangan dan Pembuatan Sistem Kontrol

Tahap ketiga yaitu merancang sistem *path tracking* robot yang digunakan untuk mengetahui perubahan posisi robot dan sistem PID untuk mengontrol kecepatan PWM pada motor DC robot.

4. Pengujian Alat

Tahap keempat yaitu melakukan pengujian alat untuk mendapatkan sistem kontrol dan *monitoring* yang dapat mengawasi dan mengontrol perubahan posisi robot. Dalam kegiatan ini dilakukan pengujian *path tracking* menggunakan metode *odometry* dengan sensor *rotary encoder* dan CMPS12, dan pengujian kontrol perubahan posisi robot.

5. Pengujian dan Pengambilan Data

Tahap kelima yaitu melakukan pengujian secara menyeluruh kerja sistem dengan mengambil data perubahan posisi robot secara *realtime*, dan hasil

pengujian kontrol dengan memerintah robot untuk menuju ke koordinat yang diinginkan.

6. Analisis Data dan Pembuatan Laporan

Tahap keenam yaitu dengan melakukan analisa pada data yang diperoleh dan membuatnya menjadi sebuah bentuk laporan hasil dari penelitian yang telah dilakukan.

7. Kesimpulan dan Saran

Tahap ketujuh yaitu dengan memberi kesimpulan dari pengujian dan analisa data serta memberikan saran untuk penelitian selanjutnya.

3.4 Alat dan Bahan Penelitian

Peralatan yang dibutuhkan dalam penelitian ini adalah :

1. *Access point*
2. Kompas prisma
3. Alat-alat ukur : Multimeter, Alat ukur jarak
4. Perkakas mekanis : Solder, Palu, Obeng, Gergaji, Gerenda, Bor
5. Perangkat lunak : Windows 10, Eagle, Arduino IDE, Microsoft Visual Studio 2010, Microsoft Acces 2016.

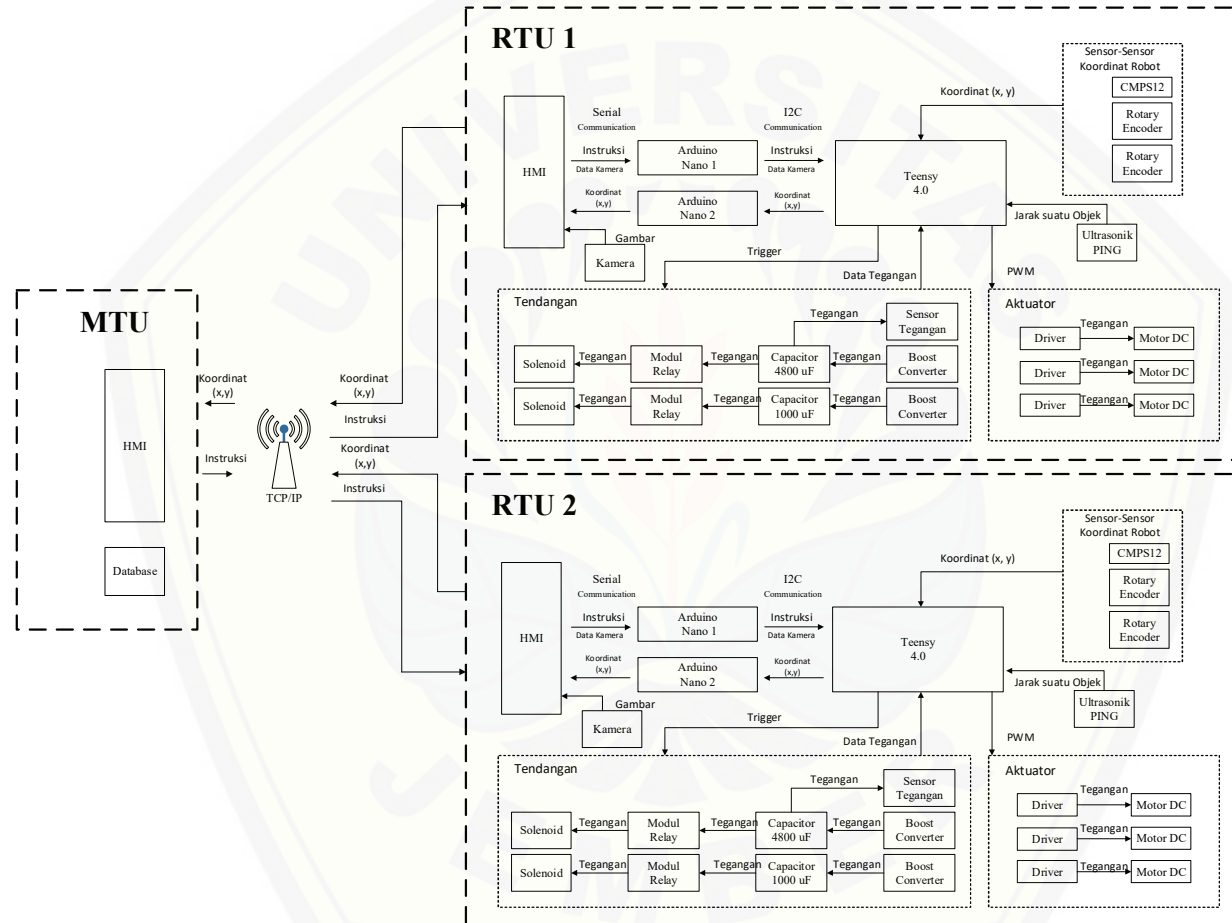
Beberapa bahan yang digunakan pada penelitian ini adalah :

1. *Personal Computer (PC)* atau Laptop
2. Mikrokontroler Teensy 4.0
3. Arduino Nano
4. *Rotary Encoder*
5. CMPS12
6. Sensor Tegangan
7. Sensor Jarak Ping
8. *Three Omnidirectional Wheeled Robot*

3.5 Blok Diagram dan *Flowchart* Sistem Kontrol dan *Monitoring*

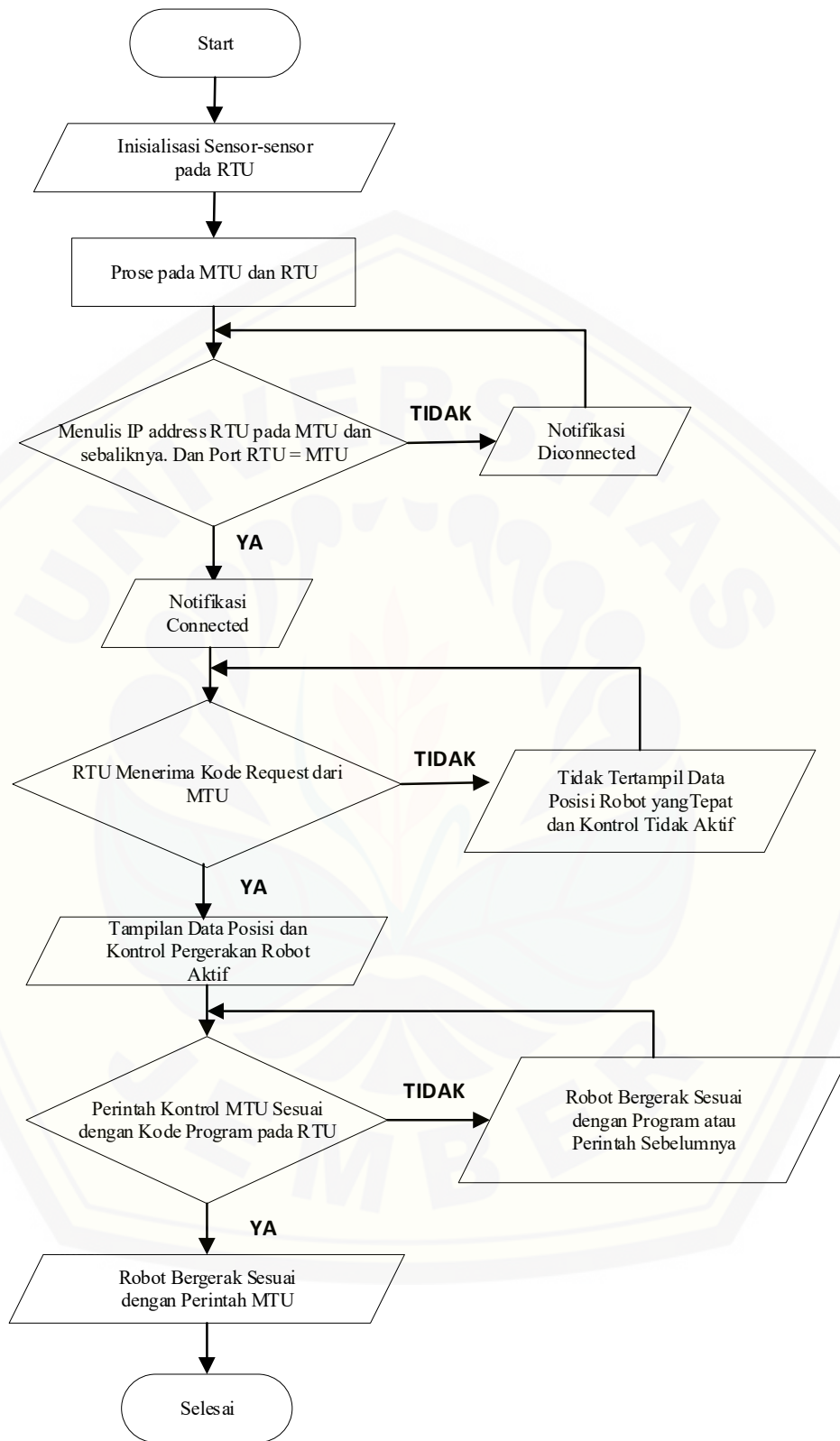
Pada penelitian ini untuk mendapatkan data dan hasil yang sesuai dengan tujuan awal maka dirancang blok diagram sistem dalam Gambar 3.2 dan *flowchart* dalam Gambar 3.3.





Gambar 3. 2 Blok Diagram Sistem Kontrol dan *Monitoring*

Sistem kontrol dan *monitoring* pada penelitian ini terdiri dari satu MTU dan dua RTU. MTU terdiri dari HMI dan *database*. HMI pada MTU berfungsi sebagai tampilan atau *monitoring* koordinat posisi dari kedua robot atau RTU secara *real time*, juga sebagai penghubung data dari maupun ke *database*, dan juga untuk mengontrol pergerakan kedua robot atau RTU. *Database* berfungsi sebagai penyimpan data perubahan posisi robot secara *realtime*. Sedangkan RTU terdiri dari HMI yang digunakan untuk antarmuka penghubung antara kontrol RTU dan MTU juga sebagai pemrosesan gambar dari sensor kamera untuk membaca bola, Teensy 4.0 sebagai pengolah data dan kontrol inti dari RTU, dua Arduino Nano dimana yang satu sebagai penerus data dari HMI RTU ke Teensy 4.0 dan Arduino Nano yang lain sebagai penerus data dari Teensy 4.0 ke HMI RTU, dua sensor rotary encoder dan CMPS12 sebagai data posisi robot secara *realtime* menggunakan metode *odometry*, sensor ultrasonik ping sebagai pembaca ada halangan atau tidak, dua kamera sebagai pembacaan sudut dan jarak bola terhadap robot dimana kamera1 sebagai kamera depan dan kamera2 sebagai kamera 360, sensor tegangan sebagai pengukur tegangan pada *capacitor bank*, tendangan bola dengan memanfaatkan solenoid dengan diameter kawat email 1,5 mm 1000 lilitan yang dialiri arus dengan tegangan 400 VDC dari kapasitor 4800 μF (tendangan keras) dan kapasitor 1000 μF (tendangan pelan) yang diisi oleh tegangan 400 volt dari tegangan 16 volt yang telah dinaikkan oleh *boost converter* sehingga menghasilkan medan magnetik untuk menggerakkan besi *plunjer* sebagai penendang bola, dan penggerak robot yang terdiri dari driver BTS7960 dan motor DC 24 volt *geared* 500 rpm.



Gambar 3. 3 Flowchart Sistem Kontrol dan Monitoring

Untuk menghubungkan antara MTU dengan RTU melalui TCP/IP yaitu dengan menuliskan IP address RTU ke MTU pada HMI dan sebaliknya, juga dengan memilih port yang sama antara MTU dengan RTU.

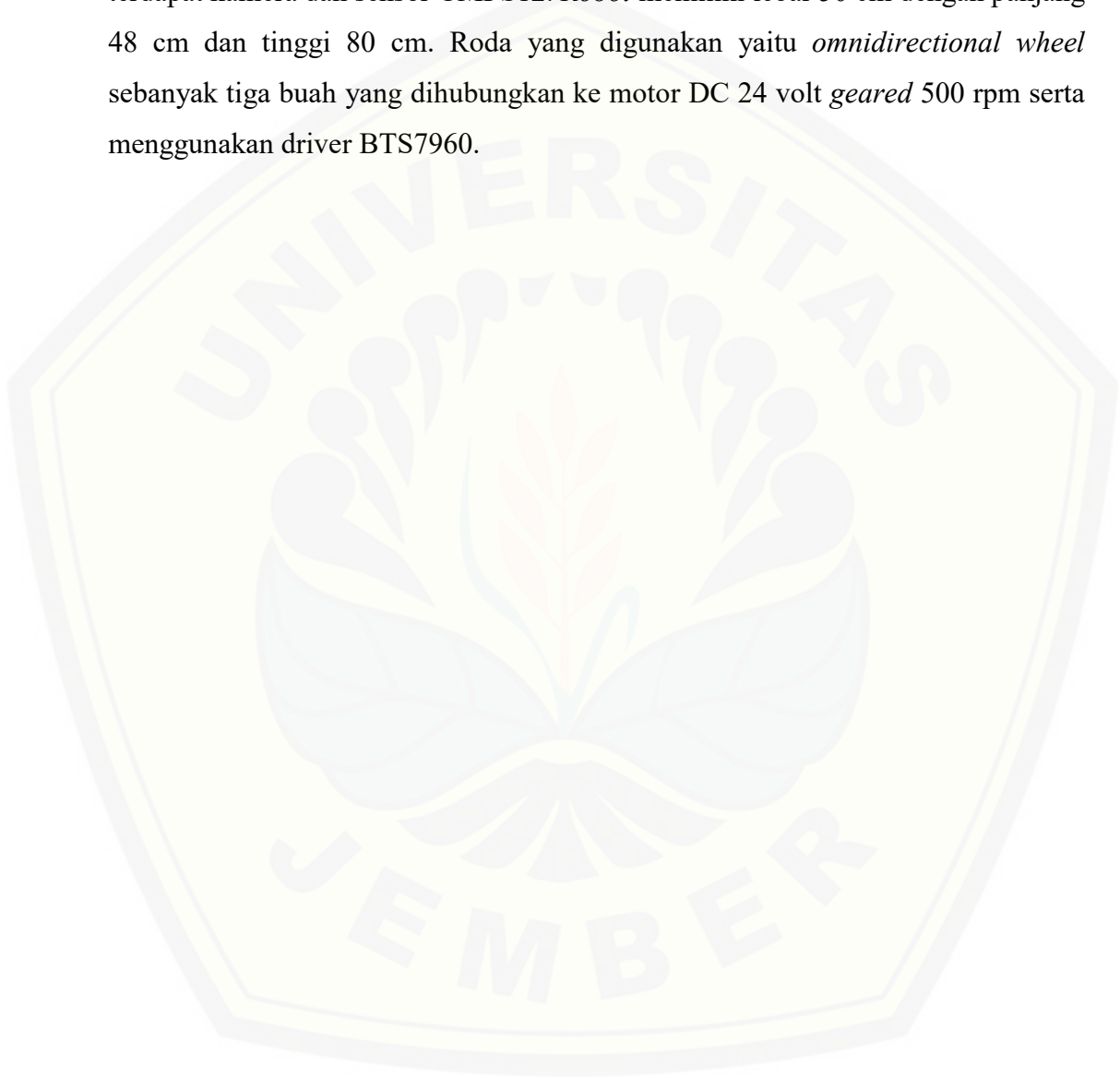
Untuk kontrol kedua robot, MTU akan mengirimkan kode instruksi ke masing-masing RTU. Lalu RTU akan menerjemahkan dan melaksanakan instruksi tersebut. Dimana instruksi tersebut akan diterjemahkan dan dilaksanakan oleh Teensy 4.0. Instruksi tersebut harus sesuai dengan program yang sudah ada pada RTU. Jika Perintah tersebut tidak sesuai maka RTU akan tetap menjalankan program atau perintah sebelumnya. Sedangkan untuk memonitoring posisi kedua RTU, MTU akan mengirim sebuah kode *request* ke masing-masing RTU, lalu masing-masing RTU akan membalas dengan mengirimkan koordinat (x,y) posisi robot dari sensor CMPS12 dan dua sensor *rotary encoder* yang diolah di Teensy 4.0 menggunakan metode *odometry*.

Berikut merupakan gambar dari *three omnidirectional wheeled robot* yang digunakan pada penelitian kali ini yang terdapat pada Gambar 3.4.



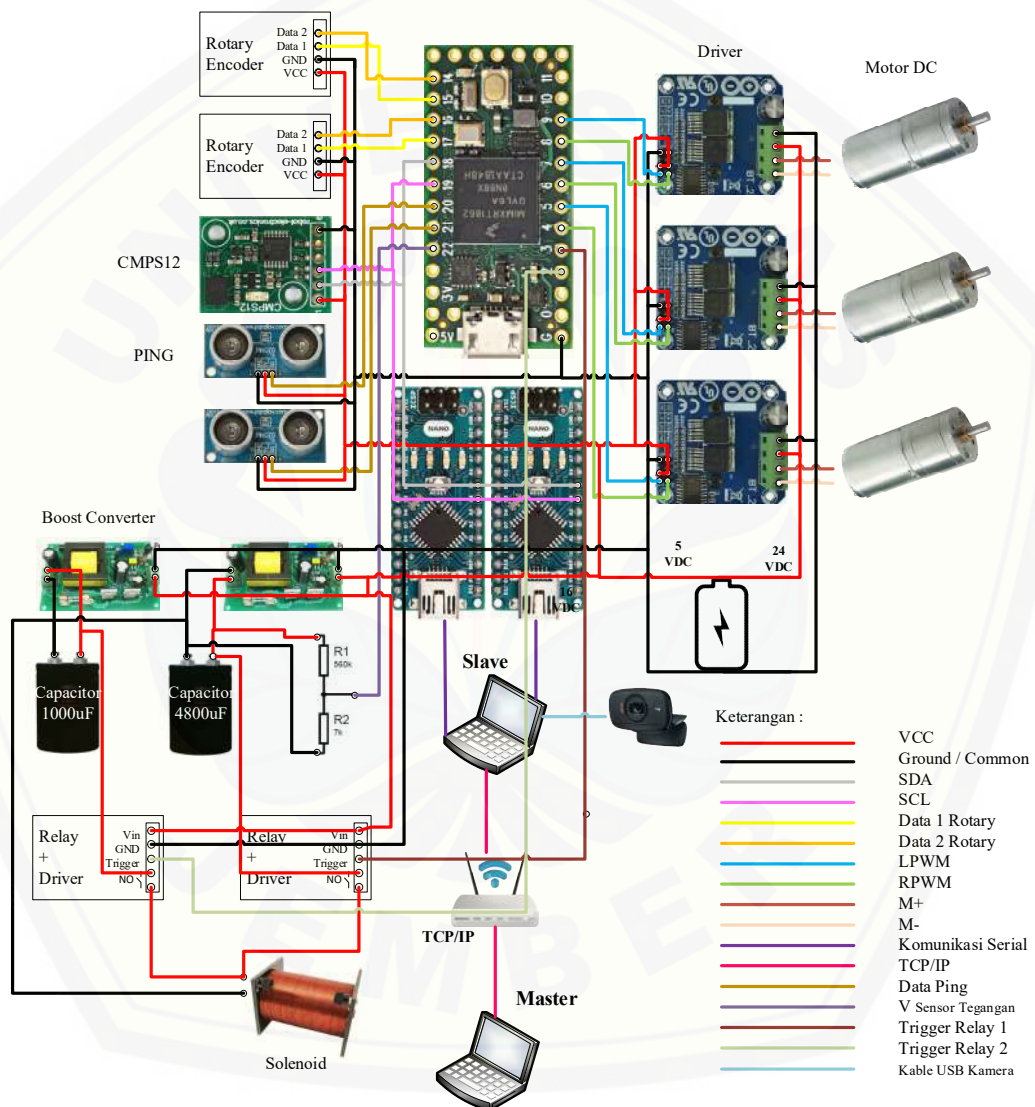
Gambar 3. 4 *Three Omnidirectional Wheeled Robot*

Badan robot dibuat menggunakan plat alumunium dengan tebal 5 mm empat tingkat. Tingkat pertama terdapat motor DC sebagai penggerak robot dan sensor *rotary encoder*, tingkat kedua terdapat rangkaian-rangkaian elektronik robot, tingkat ketiga terdapat tempat laptop dan sensor ultrasonik ping, tingkat keempat terdapat kamera dan sensor CMPS12. *Robot* memiliki lebar 50 cm dengan panjang 48 cm dan tinggi 80 cm. Roda yang digunakan yaitu *omnidirectional wheel* sebanyak tiga buah yang dihubungkan ke motor DC 24 volt *geared* 500 rpm serta menggunakan driver BTS7960.



3.6 Wiring Diagram Sistem Kontrol dan Monitoring

Wiring Diagram digunakan untuk menggambarkan sistem kontrol dan *monitoring* secara menyeluruh dan untuk mempermudah pengambilan data dan mengurangi kesalahan. *Wiring diagram* sistem kontrol dan *monitoring* yang dirancang dapat dilihat pada Gambar 3.5.



Gambar 3.5 *Wiring Diagram* Perkabelan Sistem Kontrol dan *Monitoring*

3.7 Perancangan HMI

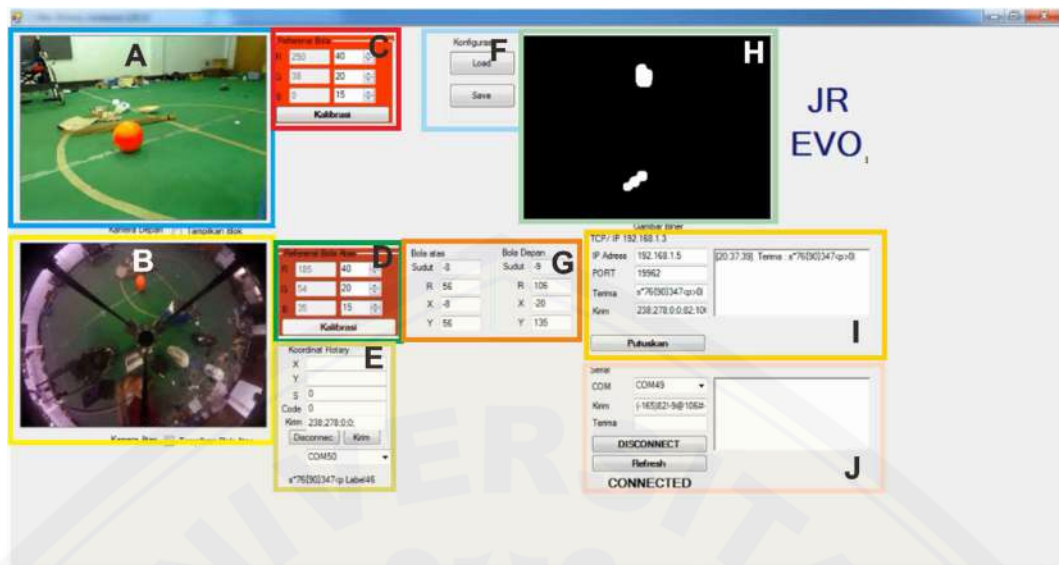
Pada penelitian ini HMI dibuat dengan menggunakan *software* Microsoft Visual Studio 2010 dengan bahasa pemrograman Visual Basic pada laptop. HMI berfungsi sebagai antarmuka yang menghubungkan antara operator atau manusia dengan sistem kontrol dan *monitoring*. HMI sendiri terdapat pada MTU dan RTU. Pada HMI MTU tombol-tombol kontrol sudah diterjemahkan menjadi bahasa manusia agar mudah dimengerti. Berikut merupakan desain HMI pada MTU dan HMI pada RTU.



Gambar 3. 6 Desain HMI pada MTU

Keterangan :

- A = Komunikasi TCP/IP dengan RTU
- B = Tampilan dan penyimpanan data ke *database*
- C = Kontrol ke RTU
- D = Tampilan posisi robot secara *real time*



Gambar 3. 7 Desain HMI pada RTU

Keterangan :

A = Tampilan kamera1 atau kamera menghadap depan

B = Tampilan kamera2 atau kamera 360 derajat

C = Kalibrasi pembacaan bola pada kamera1

D = Kalibrasi pembacaan bola pada kamera2

E = Komunikasi serial dengan Arduino Nano pembawa data dari Teensy 4.0

F = Penyimpanan dan membuka data kalibrasi bola

G = Tampilan sudut dan jarak bola dari kamera1 dan kamera2

H = Pembacaan bola biner

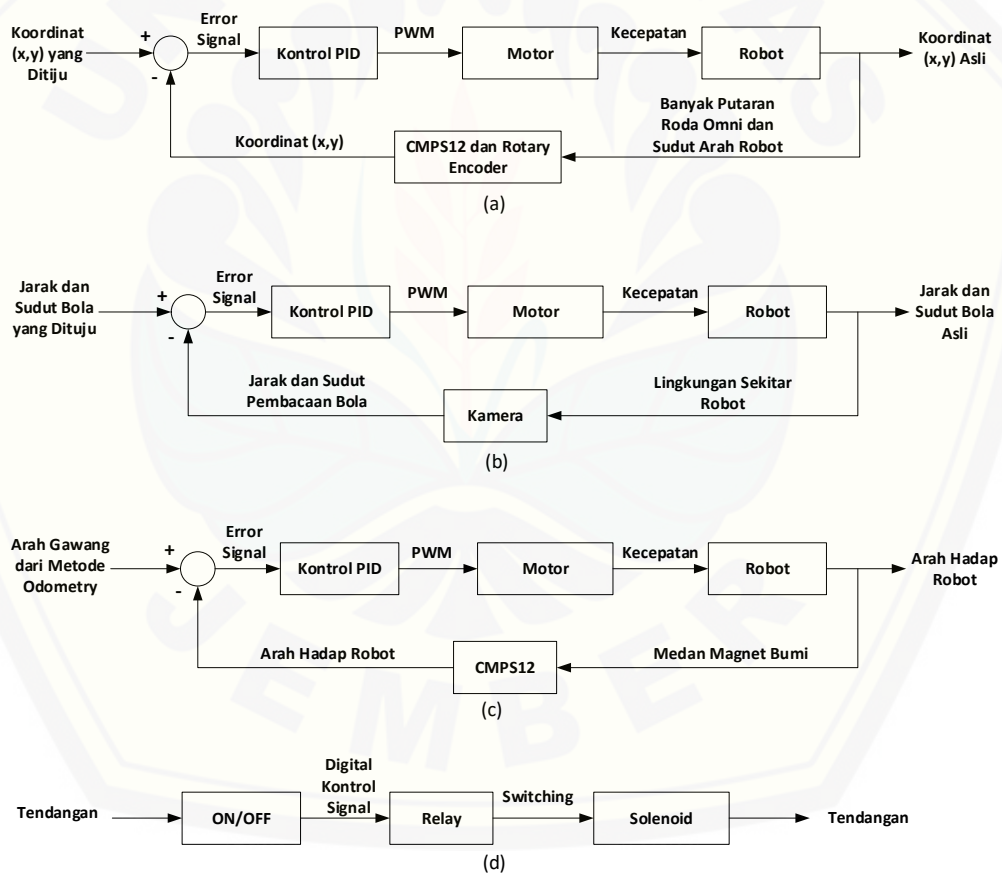
I = Komunikasi TCP/IP dengan MTU

J = Komunikasi serial dengan Arduino Nano pengirim data dan instruksi dari HMI ke Teensy 4.0

Perancangan HMI ini dibuat agar operator dapat dengan mudah menghubungkan antara MTU dengan kedua RTU, mengontrol kedua RTU dan memonitor posisi robot secara *real time* dengan tampilan gambar lapangan sesuai dengan posisi asli kedua RTU atau robot.

3.8 Perancangan Sistem Kontrol

Pada proses kontrol MTU ke RTU, MTU mengirimkan instruksi pada RTU melalui TCP/IP. Instruksi tersebut akan diterima oleh HMI pada RTU lalu instruksi tersebut dikirimkan ke Arduino Nano melalui komunikasi serial, dari Arduino Nano instruksi tersebut akan dikirim ke Teensy 4.0 melalui komunikasi I2C. Teensy 4.0 akan menerjemahkan instruksi tersebut lalu menjalankan intruksi tersebut menjadi sebuah gerakan. Terdapat 4 (empat) kontrol yang digunakan pada sistem kotrol ini, yaitu kontrol posisi robot, kontrol mengejar bola, kontrol mencari gawang lawan, dan kontrol menendang bola yang dapat dilihat pada Gambar 3.8.



Gambar 3. 8 (a) Blok Diagram Kontrol Posisi Robot, (b) Blok Diagram Kontrol Mengejar Bola, (c) Blok Diagram Kontrol Mencari Gawang Lawan, (d) Blok Diagram Kontrol Menendang Bola

Kontrol-kontrol pada Gambar 3.8 di atas akan dijalankan sesuai dengan instruksi-instruksi yang diberikan oleh MTU ke RTU. Terdapat dua instruksi kontrol yang diberikan oleh MTU ke RTU yaitu kode pertandingan KRSBI dan kode perintah agar robot dapat menjadi penyerang atau bertahan. Kode-kode ini dibuat sendiri oleh peneliti. Penjelasan kode pertandingan KRSBI dapat dilihat pada **Lampiran 1. Tabel Kode Pertandingan KRSBI.**

Ketika RTU menerima kode di atas maka RTU akan bergerak sesuai dengan arti kode di atas. Sedangkan kode perintah agar robot dapat menjadi penyerang atau bertahan akan dilaksanakan RTU ketika RTU menerima kode “s” dengan arti memulai pertandingan oleh MTU terlebih dahulu. Kode perintah agar robot dapat menjadi penyerang atau bertahan ini diberikan sesuai dengan jarak bola terhadap robot dari pengolahan citra sensor kamera. Robot yang jaraknya paling dekat ke bola akan diberi kode “p” atau penyerang dan jika robot jaraknya paling jauh ke bola akan diberi kode “b” atau bertahan.

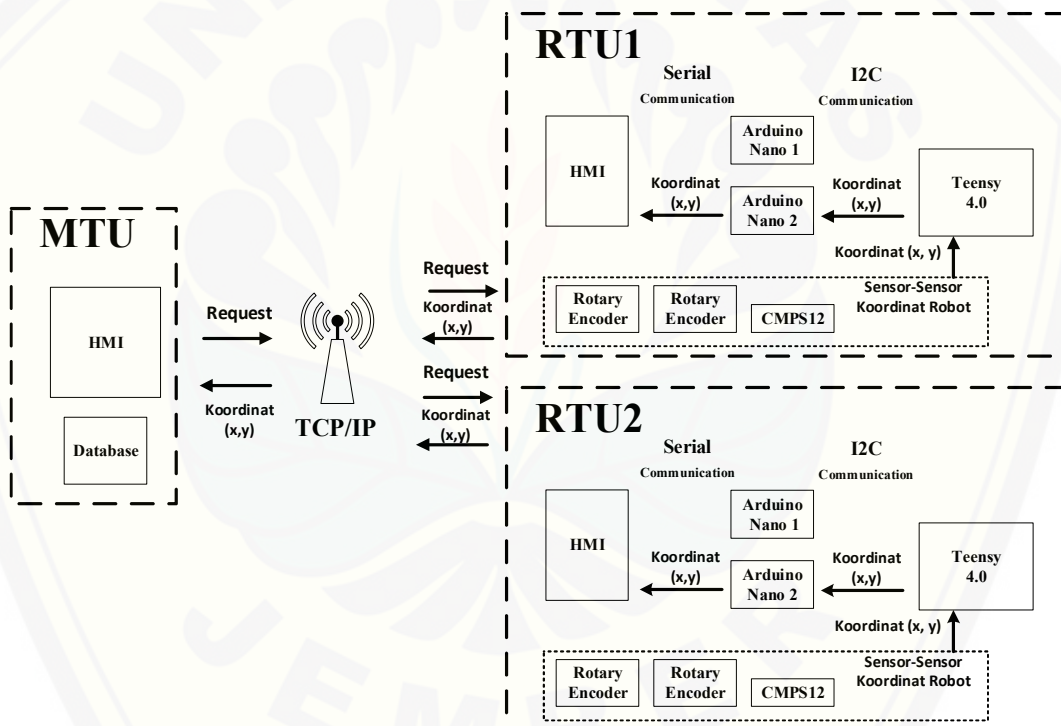
Saat kedua RTU menerima kode “s” dan menerima kode perintah agar robot dapat menjadi penyerang atau bertahan, kedua RTU atau robot akan melakukan kontrol internal seperti menggiring bola, mengejar bola, menendang bola, atau mengumpan bola sesuai dengan kondisi pada robot. Kondisi-kondisi tersebut, yaitu jarak bola terhadap tiap robot, ada atau tidaknya halangan pada tiap robot, dan jangkauan tendangan pada robot penyerang. Tindakan kedua robot terhadap kondisi-kondisi pada kedua robot dapat dilihat pada **Lampiran 2. Tabel Tindakan Robot terhadap Kondisi pada Robot.**

Pada kontrol *internal*, RTU mengontrol saat bergerak mengejar bola, menendang bola, dan menggiring bola dengan menggunakan sensor kamera dengan pemrosesan citra dengan menggunakan openCV pada Visual Studio dan sensor ultrasonik PING. Data dari pemrosesan citra pembacaan bola yaitu data jarak dan sudut bola terhadap robot. Terdapat dua kamera yang digunakan yaitu kamera 360 derajat yang dapat melihat sekeliling robot atau kamera1 dan kamera menghadap depan atau kamera2. Jika bola tidak terbaca pada kamera1 sudut yang terbaca adalah -51 dan jarak yang terbaca adalah 156. Jika bola tidak terbaca pada kamera2 sudut yang terbaca adalah -90 dan jarak yang terbaca adalah 288. Data tersebut

dikirimkan dari laptop melalui HMI pada RTU ke Arduino Nano menggunakan komunikasi serial, lalu dari Arduino Nano dikirimkan ke Teensy 4.0 melalui komunikasi I2C. Untuk sensor ultrasonik PING sendiri digunakan untuk mengetahui adanya halangan atau tidak di depan robot.

3.9 Perancangan Sistem Monitoring

Pada proses monitoring, MTU mengirimkan sebuah *request* data pada RTU. Lalu RTU membalas dengan mengirimkan data koordinat(x,y) robot. Berikut merupakan blok diagram sistem *monitoring* yang dapat dilihat pada Gambar 3.9.



Gambar 3. 9 Blok Diagram Sistem *Monitoring*

Pada RTU dibutuhkan sensor-sensor yaitu dua buah *rotary encoder* dan satu buah CMPS12 pada Teensy 4.0. Duah buah *rotary encoder* ini diletakkan secara 90 derajat antara *rotary* yang satu dengan yang lainnya, rotary encoder ini berfungsi sebagai penghitung putaran dari sebuah gerakan pada robot. Sedangkan sensor CMPS12 digunakan untuk mengetahui arah robot menghadap sehingga robot dapat

membedakan antara gerakan maju, mundur, kanan, kiri, dan lainnya. Dari sensor-sensor tersebut data yang diperoleh akan diolah menggunakan metode *odometry* sehingga menghasilkan data koordinat (x,y) robot dengan satuan milimeter. Data koordinat (x,y) tersebut diubah terlebih dahulu menjadi satuan pixel disamakan dengan satuan pixel pada tampilan lapangan pertandingan dengan cara koordinat (x,y) robot dibagi 100 dikali 8. Setelah itu koordinat (x,y) robot dikirim ke Arduino Nano melalui komunikasi I2C, dari Arduino Nano data dikirim ke HMI pada RTU melalui komunikasi serial, dari HMI pada RTU data akan dikirim ke MTU melalui TCP/IP, lalu HMI pada MTU akan menampilkan posisi robot secara *realtime* dan menyimpan data tersebut ke *database*.

3.10 Perancangan Sistem Komunikasi

Sistem komunikasi pada sistem SCADA ini dibagi menjadi beberapa bagian yaitu komunikasi antara MTU dengan RTU, komunikasi antara HMI RTU dengan Arduino Nano, dan komunikasi antara Teensy 4.0 dengan Arduino Nano.

3.10.1 Media Komunikasi

Dalam sistem kontrol dan *monitoring* pada perubahan posisi robot menggunakan komunikasi-komunikasi agar MTU dengan RTU dapat saling kirim maupun menerima data. Komunikasi yang digunakan yaitu komunikasi TCP/IP, komunikasi serial dan komunikasi I2C.

3.10.1.1 Komunikasi TCP/IP

Komunikasi TCP/IP digunakan antara HMI MTU dengan HMI RTU. Dalam komunikasi ini membutuhkan masing-masing PC (*Personal Computer*) / laptop pada setiap MTU maupun RTU dengan menggunakan software Visual Studio 2010 dengan bahasa pemrograman Visual Basic. Dari software Visual Studio 2010 tersebut dibuatlah aplikasi atau HMI komunikasi dengan protokol TCP dimana yang satu menjadi *server* sebagai MTU dan yang lain menjadi *client* sebagai RTU.

Pada HMI MTU terdapat kolom IP *address* dan *port* yang harus diisi IP *address* kedua RTU dan menyamakan *port* pada tiap RTU. Begitu pula pada HMI

tiap RTU terdapat kolom *IP address* dan *port* yang harus diisikan *IP address* MTU dan *port* yang sama pada tiap RTU pada HMI MTU. Jadi setiap RTU memiliki *port* yang berbeda.

3.10.1.2 Komunikasi Serial

Komunikasi serial digunakan untuk komunikasi antara HMI RTU dengan Arduino Nano. Arduino Nano yang digunakan sebanyak dua buah, Arduino Nano yang satu untuk mengirim data dari sensor-sensor yang telah diolah oleh Teensy 4.0, dan Arduino Nano yang lain sebagai penerima kode perintah maupun data pembacaan bola dari kamera, lalu data tersebut dikirim ke Teensy 4.0 untuk diolah dan diproses. Untuk menggunakan komunikasi serial antara HMI RTU dengan Arduino Nano yaitu dengan menyamakan besar *baud rate* dimana *baud rate* yang digunakan sebesar 115200 bps.

3.10.1.3 Komunikasi I2C

Komunikasi I2C digunakan antara Arduino Nano dan Teensy 4.0. Teensy 4.0 dijadikan sebagai *Master* agar dapat menerima maupun mengirim data ke *Slave* dan Arduino Nano sebagai *Slave* dimana Arduino Nano yang satu sebagai pengirim data ke Teensy 4.0 dan Arduino Nano yang lain sebagai penerima data dari Teensy 4.0. Karena Teensy 4.0 sebagai Master maka Teensy 4.0 yang menentukan kapan komunikasi dimulai dan diakhiri dengan mengendalikan atau menghasilkan serial *clock* dan membangkitkan sinyal START dan STOP. Untuk mengkomunikasikan Teensy 4.0 dengan Arduino Nano yaitu dengan menyamakan alamat I2C antara keduanya. Besar data yang dikirim atau diterima antara master dan slave maksimal sebesar 32 Byte.

3.10.2 Komunikasi antara MTU dengan RTU

Untuk bahasa komunikasi antara MTU dengan kedua RTU yaitu membuat sendiri sesuai keinginan peneliti. Data yang dikirimkan antara MTU dengan RTU merupakan data *string* dengan simbol-simbol sebagai pemecah data yang satu dengan yang lain. Data dan/atau instruksi yang dikirimkan dibagi menjadi dua yaitu

data dan instruksi dari MTU ke RTU dan data dari RTU ke MTU. MTU sendiri harus mengirimkan request data terlebih dahulu lalu RTU akan mengirimkan data ke MTU. Data dan instruksi yang dikirim MTU ke RTU awal-awal yaitu (data1*, setelah mendapat data dari kedua RTU maka MTU akan mengirim data dan instruksi yaitu (data1*data2{data3}data4<data5>data6|. Data1 dan data5 merupakan data bebas dimana data ini merupakan data kontrol MTU untuk mengontrol perubahan posisi robot. Sedangkan data2, data3, data4, dan data6 merupakan data pasti dari perubahan posisi robot dari sensor *rotary encoder*, CMPS12, ultrasonik ping dan kamera yang telah diolah oleh Teensy 4.0. Data-data tersebut dapat dijelaskan pada **Lampiran 3. Tabel Kode Perintah dan Data yang Dikirim dari MTU ke RTU**. Setelah MTU mengirimkan kode *request* RTU akan membalas dengan mengirimkan beberapa data yaitu data1;data2;data3;data4;data5;data6;. Data-data tersebut dapat dijelaskan pada **Lampiran 4. Tabel Kode dan Data yang Diterima oleh MTU dari RTU**.

Pada data yang diterima oleh MTU dari RTU hanya terdapat satu simbol berbeda dengan sebelumnya yaitu “ ; “ simbol ini juga berfungsi sebagai pemisah antar data-data yang dikirim.

3.10.3 Komunikasi antara HMI RTU dengan Arduino Nano

Data dan instruksi yang dikirim oleh HMI RTU ke Arduino Nano pertama-tama sebelum MTU menerima data dari RTU yaitu (data1)data2!data3@data4# dan setelah MTU mendapatkan data dari RTU maka data yang dikirim HMI RTU ke Arduino Nano menjadi (data1)data2!data3@data4#data5*data6{data7}data8>data9<data10|. Data-data tersebut dipisahkan oleh simbol-simbol yang berbeda. Data5 dan data8 merupakan data bebas dimana data ini merupakan data kontrol MTU untuk mengontrol perubahan posisi robot. Sedangkan data1, data2, data3, data4, data6, data 7, data 9, dan data10 merupakan data pasti dari perubahan posisi robot dari sensor *rotary encoder*, CMPS12, ultrasonik ping dan kamera yang telah diolah oleh Teensy 4.0. Data-data tersebut dapat dijelaskan pada **Lampiran 5. Tabel Instruksi dan Data yang Dikirim oleh HMI RTU ke Arduino Nano**. Sedangkan data yang dikirim

oleh Arduino Nano ke HMI RTU yaitu data1;data2;data3;data4; dan dapat dijelaskan pada **Lampiran 6. Tabel Data yang Dikirim dari Arduino Nano ke HMI RTU.**

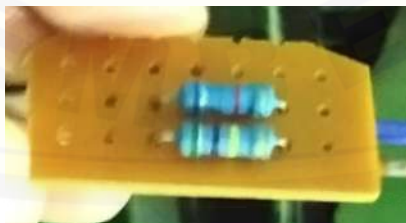
3.10.4 Komunikasi antara Arduino Nano dengan Teensy 4.0

Kode instruksi dan data yang dikirimkan oleh Arduino Nano ke Teensy 4.0. dan Data yang dikirimkan oleh Teensy 4.0. ke Arduino Nano dapat dilihat pada **Lampiran 7. Data yang Dikirimkan dari Arduino ke Teensy dan Sebaliknya melalui Komunikasi I2C.**

Besarnya data yang dikirim disesuaikan dengan kebutuhan juga kapasitas maksimal pengiriman data pada satu alamat yaitu sebesar 32 Byte. Data yang dikirim oleh Arduino ke Teensy nantinya akan diproses dan diolah oleh Teensy untuk menjalankan robot sesuai yang diinginkan. Sedangkan data yang dikirim oleh Teensy ke Arduino nantinya hanya akan diteruskan oleh Arduino ke HMI pada RTU.

3.11 Sensor Tegangan

Sensor tegangan dibuat menggunakan rangkaian pembagi tegangan. Pada perancangannya dua buah resistor dirangkai secara seri. Tegangan yang akan diukur yaitu tegangan *capasitor bank* 0 – 400 V dan dijadikan 0 – 3,3 V.



Gambar 3. 10 Sensor Tegangan

Untuk menentukan nilai dan peringkat daya resistor yang akan digunakan yaitu dengan memilih nilai resistor untuk memuat tegangan *input* (R1) yang diinginkan, mencari nilai R2 dengan perhitungan, mencari arus yang melewati

resistor, lalu peringkat daya resistor dapat diketahui. Hal ini bertujuan selain untuk menurunkan tegangan yang sesuai juga untuk menghindari kerusakan pada resistor karena beban berlebih. Pada penelitian ini R1 menggunakan resistor 560 k Ω untuk memuat tegangan input, lalu untuk nilai R2 dapat dicari menggunakan persamaan berikut:

$$V_{out} = \frac{V_{in} \times R_2}{(R_1 + R_2)}$$

Keterangan :

V_{in} : Tegangan yang akan diukur (tegangan input)

V_{out} : Tegangan yang masuk di pin analog mikrokontroler (tegangan output)

R1 : Resistansi memuat tegangan input

R2 : Resistansi pembanding

dimana V_{out} = 5 V; V_{in} = 400 V; R1 = 560 k Ω

$$3,3 = \frac{400 \times R_2}{(560k + R_2)}$$

$$3,3 \times (560k + R_2) = 400 \times R_2$$

$$2800k + 3,3R_2 = 400R_2$$

$$R_2 = \frac{2800k}{396,7} \quad R_2 = 7k \Omega$$

Dan untuk mencari peringkat daya resistor yang dibutuhkan didapatkan dari persamaan sebagai berikut:

$$V_{in} = I \times R_1$$

$$I = \frac{V_{in}}{R_1}$$

$$I = \frac{400}{560k}$$

$$I = 0,714 \text{ mA}$$

$$I = 0,714 \times 10^{-3} \text{ A}$$

peringkat daya R1:

$$P_{R1} = (V_{in} - V_{out}) \times I$$

$$P_{R1} = (400 - 3,3) \times 0,714 \times 10^{-3}$$

$$P_{R1} = 0,28 \text{ watt}$$

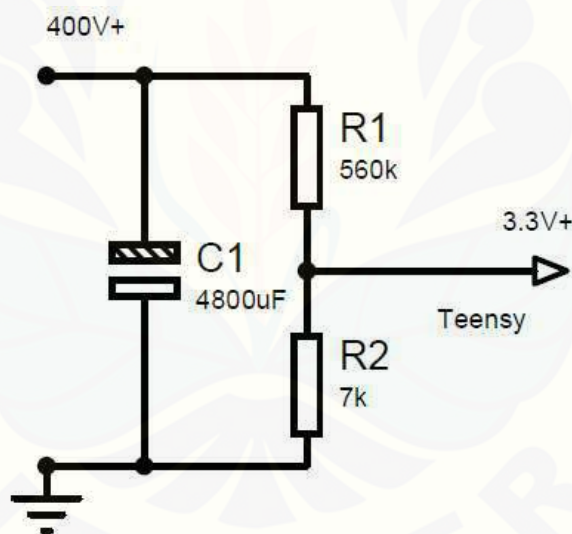
peringkat daya R2:

$$P_{R2} = (V_{out}) \times I$$

$$P_{R2} = (3,3) \times 0,714 \times 10^{-3}$$

$$P_{R2} = 0,00235 \text{ watt}$$

sehingga didapatkan R1 sebesar 560 k Ω dengan peringkat daya resistor 0,28 watt dan R2 sebesar 7k Ω dengan peringkat daya resistor 0,00235 watt. Berikut merupakan rangkaian percobaan pengukuran tegangan *capacitor bank* pada Gambar 3.10.



Gambar 3. 11 Rangkaian Pengujian Sensor Tegangan

BAB 5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dari pengujian-pengujian yang telah didapatkan pada penelitian ini, didapatkan beberapa kesimpulan sebagai berikut:

1. Sistem *monitoring* pada perubahan posisi robot Kontes Robot Sepak Bola Indonesia (KRSBI) Beroda memiliki tingkat keberhasilan yang baik. Dimana koordinat(x,y) dikirimkan secara *real time* dengan waktu pengiriman rata-rata sebesar 0,085 detik pada pengujian tanpa penyimpanan data pada *database* dan waktu pengiriman rata-rata sebesar 0,377 detik pada pengujian dengan penyimpanan data pada *database*.
2. Sistem Kontrol pada perubahan posisi robot Kontes Robot Sepak Bola Indonesia (KRSBI) Beroda memiliki tingkat keberhasilan yang baik. Dimana kode perintah atau instruksi yang dikirim dari MTU sesuai dengan kode perintah atau instruksi yang diterima oleh RTU yang memiliki waktu pengiriman rata-rata sebesar 0,081 detik dan waktu respon motor DC terhadap instruksi rata-rata sebesar 0,229 detik pada pengujian tanpa penyimpanan data posisi robot pada *database* juga pengiriman rata-rata sebesar 0,375 detik dan waktu respon motor DC terhadap instruksi rata-rata sebesar 0,52 detik pada pengujian dengan penyimpanan data posisi robot pada *database*.
3. Pada sistem kontrol, kode perintah dikirim oleh MTU ke RTU melalui komunikasi TCP/IP. Pada RTU kode instruksi tersebut pertama-tama masuk ke PC, lalu dari PC dikirimkan melalui Arduino Nano melalui komunikasi serial, lalu dari Arduino Nano dikirim melalui komunikasi I2C ke Teensy 4.0, lalu di Teensy 4.0 kode instruksi diterjemahkan dan dijalankan.
4. Pada sistem monitoring, ketika RTU sudah mendapat kode *request* berupa simbol “(“, RTU akan membalas dengan mengirim data koordinat (x,y) robot dari data sensor-sensor yang telah diolah oleh Teensy 4.0. Dari Teensy 4.0 data dikirim ke Arduino Nano melalui komunikasi I2C, lalu dari Arduino Nano data dikirim ke PC melalui komunikasi serial dan dari PC tersebut data dikirimkan ke MTU melalui komunikasi TCP/IP.

5. Penggunaan sistem kontrol pada perubahan posisi robot Kontes Robot Sepak Bola Indonesia (KRSBI) Beroda yaitu untuk mengontrol pergerakan robot berupa gerak pada posisi tertentu dan gerakan strategi seperti robot menggiring, robot mengumpan, dan robot menendang.

5.2 Saran

Pada penelitian tentang sistem kontrol dan *monitoring* pada perubahan posisi robot Kontes Robot Sepak Bola Indonesia (KRSBI) beroda, terdapat beberapa saran dari penulis dengan harapan penelitian selanjutnya dapat membuahkan hasil yang lebih maksimal, yaitu:

1. Komunikasi antara MTU dengan RTU dapat dibuat dengan jarak yang lebih jauh lagi tidak terpaku pada jaringan WIFI, yaitu dengan MTU menyambung pada jaringan kabel LAN sedangkan RTU tetap tersambung pada jaringan WIFI..
2. Penambahan *Artificial Intelligence* (AI) atau kecerdasan buatan pada kontrol robot.
3. Mekanik pada robot harus diperhatikan dengan baik sehingga robot dapat dikontrol sesuai dengan yang diinginkan. Seperti pemasangan tiga roda *omni-directional* pada robot harus dengan sudut 120 derajat antar roda dan pemasangan besi tendangan bola dengan lurus.

DAFTAR PUSTAKA

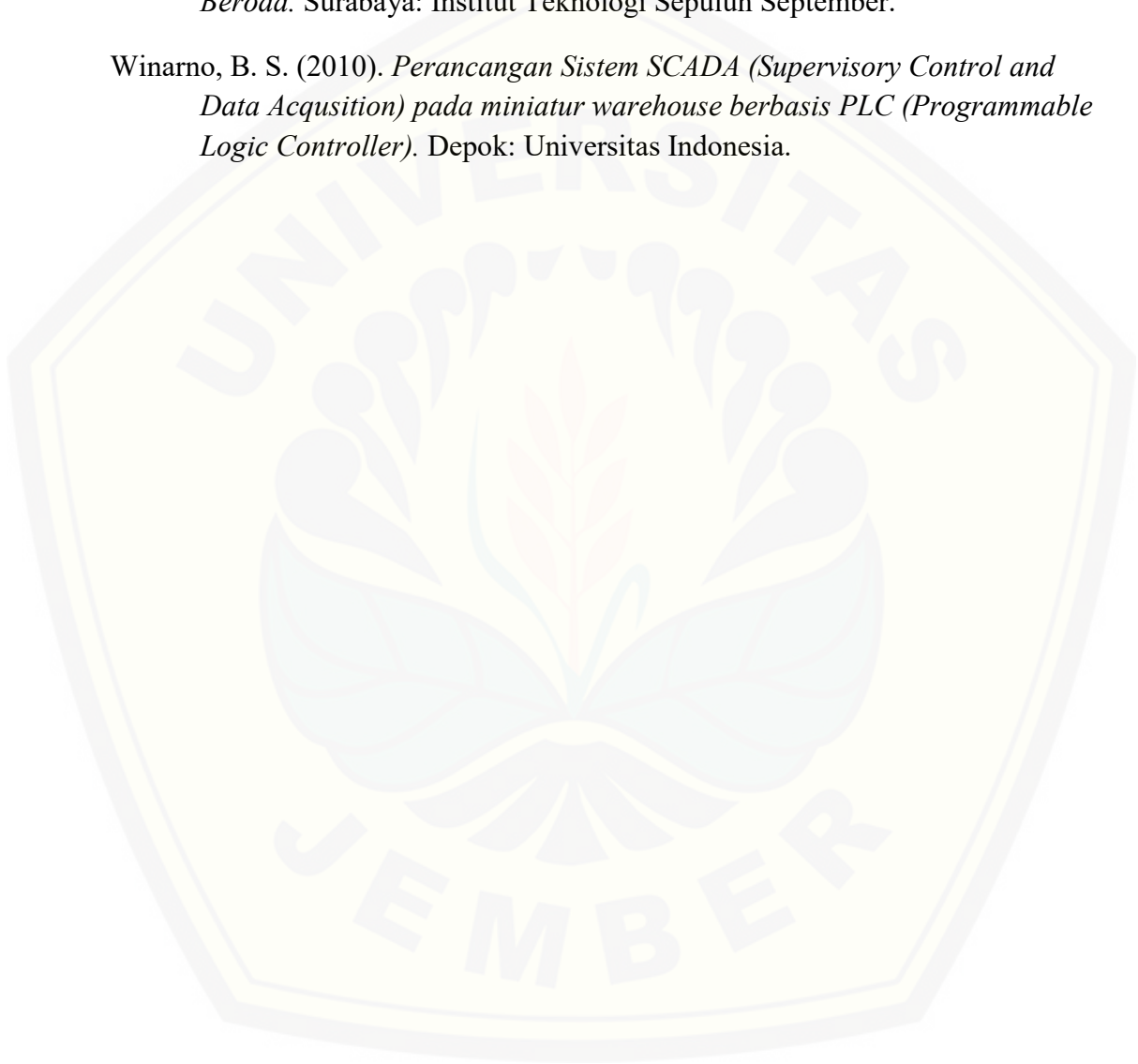
- Anita, S., Joshi, S., & Shinta, G. (2016). SCADA Implementation for Power Generator Performance and Condition Monitoring System. *International Journal of Advanced Research in Computer and Communication VOL 5*, 557-558.
- Ardhiansyah, T., Hartatik, O. T., Mustaqim, I., Naufal, M. R., Pramono, Y., & Syarifuddin, I. (2017). Pergerakan Otomatis Robot Sepak Bola Beroda Melalui Komunikasi dengan Referee Box Menggunakan Base Station. *5th Indonesian Symposium on Robotics Systems and Control*, 82-86.
- Banodin, R. (2006). *Alat Petunjuk Arah Mata Angin dan Pengukur Kecepatan Angin Berbasis Mikrokontroler AT89C51*. Semarang: Universitas Diponegoro.
- Basori, S., Nanang, S., & Rifan, M. (2014). *Implementasi Odometry pada Robot Otomatis Kontes Robot Abu Indonesia*. Malang: Universitas Brawijaya.
- Borenstein, J., & L, F. (1996). Gyrodometry A New Method for Combining Data from Gyros and Odometry in Mobile Robot. *IEEE International Conference on Robotics and Automation*, 423-428.
- CH, S. S., Agrawal, A., & R, A. K. (2020). Quanta - A Platform for Rapid Control and Monitoring of Heterogeneous Robots. *Digital Communications and Networks*.
- Chookaew, S., Howimanporn, S., & Sootkaneung, W. (2018). Implementation of PSO Based Gain-Scheduled PID and LQR for DC Motor Control Using PLC and SCADA. *2018 International Conference on Control and Robots (ICCR)*.
- Hadiyanto, R, W., & Prasetyono, S. (2015). Prototype Sistem Compressed Air Energy Storage dengan Pengaturan Valve Menggunakan Fuzzy Logic Controller. *Berkala Sainstek*.
- Isler, Y., & Sahin, S. (2013). Microcontroller-Based Robotics and SCADA Experiments. *International Journal IEE Transaction on Education*.
- Kemendikbud. (2020). *Petunjuk Pelaksanaan Kontes Robot Indonesia Tahun 2020*. Jakarta: KEMENDIKBUD.

Millah, I. S. (2017). *Implementasi Sistem SCADA pada Generator DC Penguatan Terpisah Menggunakan Arduino Mega 2560*. Jember: Universitas Jember.

Novel, R. (2009). *Analisa Perancangan Sistem SCADA pada Sistem Kelistrikan Universitas Indonesia*. Depok: Universitas Indonesia.

Rachman, A. A. (2017). *Sistem Perencanaan Rute Gerak pada Robot Sepak Bola Beroda*. Surabaya: Institut Teknologi Sepuluh September.

Winarno, B. S. (2010). *Perancangan Sistem SCADA (Supervisory Control and Data Acquisition) pada miniatur warehouse berbasis PLC (Programmable Logic Controller)*. Depok: Universitas Indonesia.



LAMPIRAN

Lampiran 1. Tabel Kode Pertandingan KRSBI

Kode	Arti Kode
l	Instruksi robot berputar searah jarum jam
j	Instruksi robot berputar berlawanan arah jarum jam
i	Instruksi robot bergerak maju
m	Instruksi robot bergerak mundur
o	Instruksi robot bergerak ke kanan
u	Instruksi robot bergerak ke kiri
e	Instruksi robot kalibrasi koordinat (x, y) sesuai dengan posisi <i>real</i>
p	Instruksi robot kalibrasi arah hadap robot terhadap gawang lawan
s	Instruksi robot memulai pertandingan
S	Instruksi robot berhenti
Q	Intruksi robot menuju ke posisi persiapan <i>corner</i> kanan gawang lawan
W	Intruksi robot menuju ke posisi persiapan <i>corner</i> kiri gawang lawan
c	Instruksi robot menuju posisi bertahan saat musuh melakukan <i>corner</i>
N	Instruksi robot menuju posisi persiapan <i>drop ball</i>
G	Instruksi robot menuju posisi persiapan <i>goal kick</i>
K	Instruksi robot menuju posisi persiapan saat <i>kick off</i>
k	Instruksi robot menuju posisi persiapan saat musuh <i>kick off</i>
g	Instruksi robot menuju posisi persiapan saat musuh <i>goal kick</i>

Lampiran 2. Tabel Tindakan Robot terhadap Kondisi pada Robot

Kondisi Robot atau RTU	Tindakan Robot atau RTU
Tidak ada robot yang mendapat bola	Robot penyerang mengejar bola dan robot bertahan melakukan <i>back-up</i> di belakang robot penyerang.
Robot penyerang mendapat bola dan tidak ada halangan di depan robot penyerang..	Robot penyerang menendang bola ke gawang lawan dan robot bertahan melakukan <i>back-up</i> di belakang robot penyerang
Robot penyerang mendapat bola dan terdapat halangan di depan robot penyerang. Robot bertahan tidak terhalang.	Robot penyerang mengumpan kepada robot bertahan
Robot penyerang mendapat bola dan terdapat halangan di depan robot penyerang. Robot bertahan juga terhalang.	Robot penyerang menggiring bola dan robot bertahan diam memberi jalan pada robot bertahan
Robot penyerang mendapat bola dan tidak terdapat halangan di depan robot penyerang tetapi jarak tendangan tidak mencukupi. Robot bertahan tidak terhalang.	Robot penyerang mengumpan kepada robot bertahan
Robot penyerang mendapat bola dan tidak terdapat halangan di depan robot penyerang tetapi jarak tendangan tidak mencukupi. Robot bertahan juga terhalang.	Robot penyerang menggiring bola dan robot bertahan diam memberi jalan pada robot bertahan

Lampiran 3. Tabel Kode Perintah dan Data yang Dikirim dari MTU ke RTU

Instruksi atau Data	Deskripsi
(Merupakan kode <i>request</i> ke RTU. Sehingga RTU dapat membalas dengan mengirimkan data koordinat(x,y) posisi robot dan data yang lainnya.
data1*	Merupakan kode standart dalam pertandingan KRSBI yang dikirim oleh wasit dan kode yang dibuat sendiri oleh peneliti. Kode ini bisa menuruti kemauan wasit atau dapat diubah menurut MTU.
data2{	Merupakan data koordinat x dari sensor <i>rotary encoder</i> dan CMPS12 yang telah diolah menggunakan metode <i>odometry</i> pada robot atau RTU kawan atau Robot kawan yang telah diubah nilainya sebanding dengan nilai pixel gambar lapangan pada MTU
data3}	Merupakan data koordinat y dari sensor <i>rotary encoder</i> dan CMPS12 yang diolah menggunakan metode <i>odometry</i> pada robot atau RTU kawan atau Robot kawan yang telah diubah nilainya sebanding dengan nilai pixel gambar lapangan pada MTU
data4<	Merupakan data sudut arah hadap robot atau RTU kawan dari pembacaan sensor CMPS12
data5>	Merupakan kode perintah agar robot dapat menjadi penyerang atau bertahan sehingga kedua robot tidak saling bertabrakan saat mengambil bola dan juga dapat digunakan sebagai strategi permainan
data6	Merupakan kode kondisi robot kawan atau RTU kawan dari olahan sensor <i>rotary encoder</i> , CMPS12, ultrasonik ping, sensor tegangan, dan kamera. Kode ini dapat berubah arti sesuai dengan data5 robot tersebut sebagai penyerang atau bertahan. Berikut merupakan kode-kode beserta artinya saat robot kawan sebagai penyerang atau saat data5 adalah “p” sebagai berikut. 0 : Merebut bola atau menendang langsung bola ke gawang 1 : Menggiring Bola saat didepan robot penyerang terdapat halangan musuh dan robot bertahan terhalang oleh musuh ketika posisi siap menerima bola dan/atau kapasitas jarak tendangan tidak memungkinkan untuk masuk ke gawang lawan 2 : Posisi Siap Mengumpan saat terdapat halangan di depan robot penyerang. Atau kapasitas jarak tendangan tidak memungkinkan untuk masuk ke gawang lawan. 3 : Siap Mengumpan saat robot bertahan siap menerima bola dan tidak terdapat halangan

Sedangkan saat robot kawan sebagai bertahan atau kode data5 adalah “b”, kode-kode beserta artinya sebagai berikut.

0 : Bertahan, saat robot penyerang mengejar bola, menendang bola, dan menggiring

1 : Posisi siap menerima umpan bola ketika robot penyerang mau mengumpan bola dan robot bertahan tidak dihalangi robot musuh dengan sensor jarak ping.

2 : Siap menerima bola saat sudut bola tepat berada di depan robot bertahan.

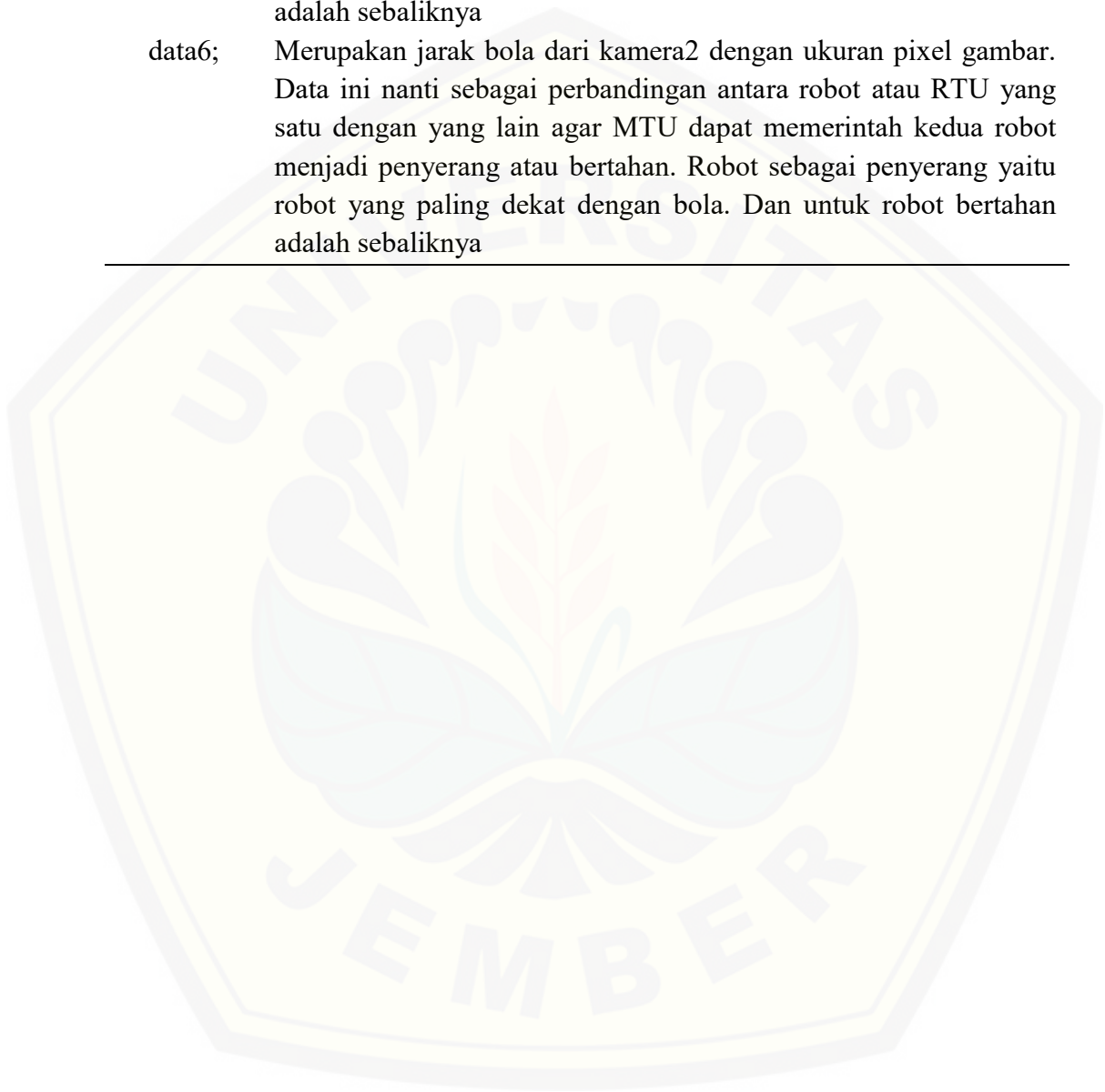
3 : Robot terhalang oleh musuh saat posisi siap untuk menerima umpan bola



Lampiran 4. Tabel Kode dan Data yang Diterima oleh MTU dari RTU

Instruksi atau Data	Deskripsi
data1;	Merupakan data koordinat x dari sensor <i>rotary encoder</i> dan CMPS12 yang telah diolah menggunakan metode <i>odometry</i> pada robot atau RTU yang telah diubah nilainya sebanding dengan nilai pixel gambar lapangan pada MTU.
data2;	Merupakan data koordinat y dari sensor <i>rotary encoder</i> dan CMPS12 yang telah diolah menggunakan metode <i>odometry</i> pada robot atau RTU yang telah diubah nilainya sebanding dengan nilai pixel gambar lapangan pada MTU
data3;	Merupakan data sudut arah hadap robot atau RTU dari pembacaan sensor CMPS12
data4;	<p>Merupakan kode kondisi robot dari olahan sensor <i>rotary encoder</i>, CMPS12, ultrasonik ping, sensor tegangan, dan kamera. Kode ini dapat berubah arti sesuai dengan data5 yang dikirim MTU ke RTU robot tersebut sebagai penyerang atau bertahan. Berikut merupakan kode-kode beserta artinya saat robot sebagai penyerang atau saat data5 adalah “p” sebagai berikut.</p> <p>0 : Merebut bola atau menendang langsung bola ke gawang 1 : Menggiring Bola saat didepan robot penyerang terdapat halangan musuh dan robot bertahan terhalang oleh musuh ketika posisi siap menerima bola dan/atau kapasitas jarak tendangan tidak memungkinkan untuk masuk ke gawang lawan 2 : Posisi Siap Mengumpan saat terdapat halangan di depan robot penyerang. Atau kapasitas jarak tendangan tidak memungkinkan untuk masuk ke gawang lawan. 3 : Siap Mengumpan saat robot bertahan siap menerima bola dan tidak terdapat halangan</p> <p>Sedangkan saat robot sebagai bertahan atau kode data5 adalah “b”, kode-kode beserta artinya sebagai berikut.</p> <p>0 : Bertahan, saat robot penyerang mengejar bola, menendang bola, dan menggiring 1 : Posisi siap menerima umpan bola ketika robot penyerang mau mengumpan bola dan robot bertahan tidak dihalangi robot musuh dengan sensor jarak ping. 2 : Siap menerima bola saat sudut bola tepat berada di depan robot bertahan. 3 : Robot terhalang oleh musuh saat posisi siap untuk menerima umpan bola</p>

- data5; Merupakan jarak bola dari kamera1 dengan ukuran pixel gambar. Data ini nanti sebagai perbandingan antara robot atau RTU yang satu dengan yang lain agar MTU dapat memerintah kedua robot menjadi penyerang atau bertahan. Robot sebagai penyerang yaitu robot yang paling dekat dengan bola dan untuk robot bertahan adalah sebaliknya
- data6; Merupakan jarak bola dari kamera2 dengan ukuran pixel gambar. Data ini nanti sebagai perbandingan antara robot atau RTU yang satu dengan yang lain agar MTU dapat memerintah kedua robot menjadi penyerang atau bertahan. Robot sebagai penyerang yaitu robot yang paling dekat dengan bola. Dan untuk robot bertahan adalah sebaliknya
-



Lampiran 5. Tabel Instruksi dan Data yang Dikirim oleh HMI RTU ke Arduino Nano

Kode Perintah atau Data	Deskripsi
(Merupakan kode <i>request</i> ke RTU. Sehingga RTU dapat membalas dengan mengirimkan data koordinat(x,y) posisi robot dan data yang lainnya.
data1)	Merupakan sudut pembacaan bola dari kamera1 dengan satuan pixel.
data2!	Merupakan jarak pembacaan bola dari kamera1 dengan satuan pixel.
data3@	Merupakan sudut pembacaan bola dari kamera2 dengan satuan pixel.
data4#	Merupakan jarak pembacaan bola dari kamera2 dengan satuan pixel.
data5*	Merupakan kode standart dalam pertandingan KRSBI yang dikirim oleh wasit dan kode yang dibuat sendiri oleh peneliti. Kode ini bisa menuruti kemauan wasit atau dapat diubah menurut MTU.
data6{	Merupakan data koordinat x dari sensor <i>rotary encoder</i> dan CMPS12 yang telah diolah menggunakan metode <i>odometry</i> pada robot atau RTU kawan atau Robot kawan yang telah diubah nilainya sebanding dengan nilai pixel gambar lapangan pada MTU.
data7}	Merupakan data koordinat y dari sensor <i>rotary encoder</i> dan CMPS12 yang telah diolah menggunakan metode <i>odometry</i> pada robot atau RTU kawan atau robot EVI yang telah diubah nilainya sebanding dengan nilai pixel gambar lapangan pada MTU.
data8>	Merupakan kode kondisi dari olahan sensor <i>rotary encoder</i> , CMPS12, ultrasonik ping, sensor tegangan, dan kamera pada robot kawan. Kode ini dapat berubah arti sesuai dengan data5 robot tersebut sebagai penyerang atau bertahan. Berikut merupakan kode-kode beserta artinya saat robot kawan sebagai penyerang atau saat data5 adalah “p” sebagai berikut. 0 : Merebut bola atau menendang langsung bola ke gawang

1 : Menggiring Bola saat didepan robot penyerang terdapat halangan musuh dan robot bertahan terhalang oleh musuh ketika posisi siap menerima bola dan/atau kapasitas jarak tendangan tidak memungkinkan untuk masuk ke gawang lawan

2 : Posisi Siap Mengumpan saat terdapat halangan di depan robot penyerang. Atau kapasitas jarak tendangan tidak memungkinkan untuk masuk ke gawang lawan.

3 : Siap Mengumpan saat robot bertahan siap menerima bola dan tidak terdapat halangan Sedangkan saat robot kawan sebagai bertahan atau kode data5 adalah “b”, kode-kode beserta artinya sebagai berikut.

0 : Bertahan, saat robot penyerang mengejar bola, menendang bola, dan menggiring

1 : Posisi siap menerima umpan bola ketika robot penyerang mau mengumpan bola dan robot bertahan tidak dihalangi robot musuh dengan sensor jarak ping.

2 : Siap menerima bola saat sudut bola tepat berada di depan robot bertahan.

3 : Robot terhalang oleh musuh saat posisi siap untuk menerima umpan bola

data9< Merupakan data sudut arah hadap robot atau RTU kawan dari pembacaan sensor CMPS12.

data10| Merupakan kode perintah agar robot dapat menjadi penyerang atau bertahan sehingga kedua robot tidak saling bertabrakan saat mengambil bola dan juga dapat digunakan sebagai strategi permainan.

Lampiran 6. Tabel Data yang Dikirim dari Arduino Nano ke HMI RTU

Data yang Dikirim	Deskripsi
data1;	Merupakan data koordinat x dari sensor <i>rotary encoder</i> dan CMPS12 yang telah diolah menggunakan metode <i>odometry</i> pada robot atau RTU yang telah diubah nilainya sebanding dengan nilai pixel gambar lapangan pada MTU
data2;	Merupakan data koordinat y dari sensor <i>rotary encoder</i> dan CMPS12 yang telah diolah menggunakan metode <i>odometry</i> pada robot atau RTU yang telah diubah nilainya sebanding dengan nilai pixel gambar lapangan pada MTU.
data3;	Merupakan data sudut arah hadap robot atau RTU dari pembacaan sensor CMPS12
data4;	<p>Merupakan kode kondisi robot dari olahan sensor <i>rotary encoder</i>, CMPS12, ultrasonik ping, sensor tegangan, dan kamera. Kode ini dapat berubah arti sesuai dengan data5 robot tersebut sebagai penyerang atau bertahan. Berikut merupakan kode-kode beserta artinya saat robot kawan sebagai penyerang atau saat data5 adalah “p” sebagai berikut.</p> <p>0 : Merebut bola atau menendang langsung bola ke gawang</p> <p>1 : Menggiring Bola saat didepan robot penyerang terdapat halangan musuh dan robot bertahan terhalang oleh musuh ketika posisi siap menerima bola dan/atau kapasitas jarak tendangan tidak memungkinkan untuk masuk ke gawang lawan</p> <p>2 : Posisi Siap Mengumpan saat terdapat halangan di depan robot penyerang. Atau kapasitas jarak tendangan tidak memungkinkan untuk masuk ke gawang lawan.</p> <p>3 : Siap Mengumpan saat robot bertahan siap menerima bola dan tidak terdapat halangan</p> <p>Sedangkan saat robot kawan sebagai bertahan atau kode data5 adalah “b”, kode-kode beserta artinya sebagai berikut.</p> <p>0 : Bertahan, saat robot penyerang mengejar bola, menendang bola, dan menggiring</p>

1 : Posisi siap menerima umpan bola ketika robot penyerang mau mengumpan bola dan robot bertahan tidak dihalangi robot musuh dengan sensor jarak ping.

2 : Siap menerima bola saat sudut bola tepat berada di depan robot bertahan.

3 : Robot terhalang oleh musuh saat posisi siap untuk menerima umpan bola



Lampiran 7. Data yang Dikirimkan dari Arduino ke Teensy dan Sebaliknya melalui Komunikasi I2C

Dikirim Arduino ke Teensy				Dikirim Teensy ke Arduino	
Data yang Dikirim			Byte (Max)	Data yang Dikirim	
				Byte (Max)	
Jarak	bola	dari	2 Byte	Koordinat (x, y) robot	10 Byte
pembacaan kamera1				dari sensor <i>rotary encoder</i> dan CMPS12 yang diolah menggunakan metode <i>odometry</i>	
Sudut	bola	dari	2 Byte	Kode kondisi robot	1 Byte
pembacaan kamera2				dari sensor <i>rotary encoder</i> , CMPS12, ultrasonik ping, dan kamera	
Jarak	bola	dari	2 Byte	Sudut hadap robot dari	2 Byte
pembacaan kamera2				pembacaan sensor CMPS12	
Sudut	bola	dari	2 Byte		
pembacaan kamera2					
Merupakan kode standart dalam pertandingan KRSBI yang dikirim oleh wasit dan kode yang dibuat sendiri oleh peneliti. Kode ini bisa menuruti kemauan wasit atau dapat diubah menurut MTU.			1 Byte		
Koordinat (x, y) dari robot atau RTU kawan dari sensor <i>rotary encoder</i> dan CMPS12 yang diolah menggunakan metode <i>odometry</i>			10 Byte		

Sudut robot atau RTU kawan dari sensor CMPS12	2 Byte
Kode perintah robot sebagai penyerang atau bertahan	1 Byte
Kode kondisi robot atau RTU kawan dari olahan sensor <i>rotary encoder</i> , CMPS12, ultrasonik ping, dan kamera. Kode ini dapat berubah arti sesuai dengan data5 robot tersebut sebagai penyerang atau bertahan. Berikut merupakan kode-kode beserta artinya saat robot kawan sebagai penyerang atau saat data5 adalah “p” sebagai berikut. 0 : Merebut bola atau menendang langsung bola ke gawang 1 : Menggiring Bola saat didepan robot penyerang terdapat halangan musuh dan robot bertahan terhalang oleh musuh ketika posisi siap menerima bola dan/atau kapasitas jarak tendangan tidak memungkinkan untuk masuk ke gawang lawan 2 : Posisi Siap Mengumpan saat terdapat halangan di depan robot penyerang. Atau kapasitas jarak tendangan tidak	1 Byte

memungkinkan untuk masuk ke gawang lawan.

3 : Siap Mengumpan saat robot bertahan siap menerima bola dan tidak terdapat halangan

Sedangkan saat robot kawan sebagai bertahan atau kode data5 adalah "b", kode-kode beserta artinya sebagai berikut.

0 : Bertahan, saat robot penyerang mengejar bola, menendang bola, dan menggiring

1 : Posisi siap menerima umpan bola ketika robot penyerang mau mengumpan bola dan robot bertahan tidak dihalangi robot musuh dengan sensor jarak ping.

2 : Siap menerima bola saat sudut bola tepat berada di depan robot bertahan.

3 : Robot terhalang oleh musuh saat posisi siap untuk menerima umpan bola

Lampiran 8. Tabel Data Hasil Kontrol Otomatis

No.	Kondisi Robot	Kode Perintah MTU	Tindakan Robot
1.	Jarak bola pada Robot1 lebih dekat dari pada jarak bola pada Robot2.	RTU1 : (s*161{280}296<p>0 RTU2 : (s*281{336}150	Robot1 mengejar bola dan Robot2 bertahan di belakang Robot1
2.	Jarak bola pada Robot1 lebih dekat dari pada jarak bola pada Robot2.	RTU1 : (s*299{342}63<p>0 RTU2 : (s*179{398}120	Robot1 mengejar bola dan Robot2 bertahan di belakang Robot1
3.	Robot1 mendapatkan bola, jarak tendangan mencukupi, dan tidak ada halangan didepan Robot1.	RTU1 : (s*176{132}295<p>0 RTU2 : (s*297{188}60	Robot1 menendang bola ke gawang lawan dan Robot2 bertahan di belakang Robot1
4.	Jarak bola pada Robot2 lebih dekat dari pada jarak bola pada Robot1	RTU1 : (s*124{181}3420 RTU2 : (s*243{125}64<p>0	Robot2 mengejar bola dan Robot1 bertahan di belakang Robot1
5.	Robot2 mendapatkan bola, jarak tendangan mencukupi, dan tidak ada halangan didepan Robot2.	RTU1 : (s*350{422}200 RTU2 : (s*230{366}296<p>0	Robot2 menendang bola ke gawang lawan dan Robot1 bertahan di belakang Robot2
6.	Robot1 mendapatkan bola, jarak tendangan mencukupi, dan terdapat halangan di depan Robot1. Di depan Robot2 tidak ada halangan.	RTU1 : (s*308{366}124<p>2 RTU2 : (s*188{286}3043	Robot1 mengumpan bola ke Robot2
7.	Robot1 mendapatkan bola, jarak tendangan tidak mencukupi, dan tidak terdapat halangan di depan Robot1 maupun Robot2 .	RTU1 : (s*180{495}237<p>2 RTU2 : (s*300{415}563	Robot1 mengumpan bola ke Robot2

8.	Robot1 mendapatkan bola, jarak tendangan mencukupi dan terdapat halangan di depan Robot1. Di depan Robot2 tidak ada halangan.	RTU1 : (s*268{542}123<p>2 RTU2 : (s*148{462}3023	Robot1 mengumpan bola ke Robot2
9.	Robot2 mendapatkan bola, jarak tendangan tidak mencukupi, dan tidak terdapat halangan di depan Robot2 maupun Robot1.	RTU1 : (s*358{333}563 RTU2 : (s*238{413}236<p>2	Robot2 mengumpan bola ke Robot1
10.	Robot2 mendapatkan bola, jarak tendangan mencukupi dan terdapat halangan di depan Robot2. Di depan Robot1 tidak ada halangan.	RTU1 : (s*265{378}553 RTU2 : (s*145{458}237<p>2	Robot2 mengumpan bola ke Robot1
11.	Robot1 mendapatkan bola, jarak tendangan mencukupi, dan terdapat halangan di depan Robot1 maupun Robot2.	RTU1 : (s*318{366}125<p>3 RTU2 : (s*198{286}3031	Robot1 menggiring bola dan Robot2 memberi jalan untuk Robot1
12.	Robot1 mendapatkan bola, jarak tendangan tidak mencukupi, dan tidak terdapat halangan di depan Robot1. Di depan Robot2 terdapat halangan.	RTU1 : (s*185{495}236<p>3 RTU2 : (s*305{415}561	Robot1 menggiring bola dan Robot2 memberi jalan untuk Robot1
13.	Robot1 mendapatkan bola, jarak tendangan mencukupi dan terdapat halangan di	RTU1 : (s*304{543}124<p>3 RTU2 : (s*185{463}3041	Robot1 menggiring bola dan Robot2 memberi jalan untuk Robot1

	depan Robot1 maupun Robot2		
14.	Robot2 mendapatkan bola, jarak tendangan tidak mencukupi, dan tidak terdapat halangan di depan Robot2. Di depan Robot1 terdapat halangan.	RTU1 : (s*371{334}551 RTU2 : (s*251{413}237<p>3	Robot2 menggiring bola dan Robot1 memberi jalan untuk Robot2
15	Robot2 mendapatkan bola, jarak tendangan mencukupi dan terdapat halangan di depan Robot2 maupun Robot1	RTU1 : (s*283{377}571 RTU2 : (s*163{458}237<p>3	Robot2 menggiring bola dan Robot1 memberi jalan untuk Robot2

Lampiran 9. Program MTU

```

Public Class baseStation
    Dim XMouse As Integer
    Dim YMouse As Integer
    Dim X1 As Integer
    Dim Y1 As Integer
    Dim X2 As Integer
    Dim Y2 As Integer
    Dim xCoordinate As Integer
    Dim yCoordinate As Integer
    Dim kirimrobot1 As String
    Dim kirimrobot1 As String
    Dim kirimrobot2 As String
    Dim kirimrobot2 As String
    Dim bolaatas1 As Integer
    Dim boladepan1 As Integer
    Dim bolaatas2 As Integer
    Dim boladepan2 As Integer
    Dim code2b As String
    Dim code2a As String
    Dim hilangatas As Integer
    Dim hilangdepan As Integer
    Dim ts As Integer = 0
    Dim R1X As Integer
    Dim R1Y As Integer
    Dim R2X As Integer
    Dim R2Y As Integer

    Private stopwatch As New Stopwatch

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the
'DatabasePosisiRobotDataSet.PosisiRobot' table. You can move, or remove it,
as needed.

Me.PosisiRobotTableAdapter.Fill(Me.DatabasePosisiRobotDataSet.PosisiRobot)
        Me.Text = "Base Station " & soket.LocalIP
        soket2.LocalPort = TextBox6.Text
        soket3.LocalPort = TextBox15.Text
        soket4.LocalPort = TextBox16.Text
        soket2.Listen()
        soket3.Listen()
        soket4.Listen()

End Sub

'=====
Private Sub soket2_CloseEvent(ByVal sender As Object, ByVal e As
System.EventArgs) Handles soket2.CloseEvent
    Label19.Text = "Disc"
    soket2.Close()
    soket2.Listen()
    Button2.Enabled = False

```

```

End Sub
Private Sub soket2_ConnectionRequest(ByVal sender As Object, ByVal e As
AxMSWinsockLib.DMSWinsockControlEvents_ConnectionRequestEvent) Handles
soket2.ConnectionRequest
    soket2.Close()
    If soket2.RemoteHostIP = TextBox5.Text Then
        soket2.Accept(e.requestID)
        Label19.Text = "OK"
        Button2.Enabled = True

    Else
        soket2.Close()
        soket2.Listen()
        Button2.Enabled = False

    End If
End Sub
Private Sub soket2_DataArrival(ByVal sender As Object, ByVal e As
AxMSWinsockLib.DMSWinsockControlEvents_DataArrivalEvent) Handles
soket2.DataArrival
    Dim sData As Byte
    Dim terima As String
    Dim jmlBuffer As Integer
    terima = ""
    jmlBuffer = e.bytesTotal

    While jmlBuffer > 0
        soket2.GetData(sData)
        terima = terima & Convert.ToChar(sData)
        jmlBuffer = jmlBuffer - 1
    End While

    TextBox8.Text = terima
    Try
        Dim data As String() = terima.Split(";") 'comment
        'TextBox22.Text = (-data(0)) + 63 + 480 - 30
        'TextBox19.Text = (-data(1)) + 63 + 720 - 30
        TextBox22.Text = data(0) + 33
        TextBox19.Text = data(1) + 33
        R1X = data(0)
        R1Y = data(1)
        bolaatas1 = data(4)
        boladepan1 = data(5)
        kirimrobot1 = data(2) + "<" + code2a + ">" + data(3) + "|"
        kirimrobo2 = data(0) + "{" + data(1) + "}" + data(2) + "<" +
code2a + ">" + data(3) + "|"

    Catch ex As Exception

    End Try

    hilangatas = 205
    hilangdepan = 288

```

```

If bolaatas1 = 196 Then
    bolaatas1 = 205
End If

If bolaatas1 = bolaatas2 Then
    'If boladepan1 = hilangdepan And boladepan2 = hilangdepan Then
    '    code2a = "t"
    '    code2b = "t"
    If boladepan1 < boladepan2 Then
        code2a = "b"
        code2b = "p"
    ElseIf boladepan1 > boladepan2 Then
        code2a = "p"
        code2b = "b"
    End If
Else
    If bolaatas1 < bolaatas2 Then
        code2a = "b"
        code2b = "p"
    ElseIf bolaatas1 > bolaatas2 Then
        code2a = "p"
        code2b = "b"
    End If
End If

TextBox18.Text = bolaatas1
TextBox23.Text = bolaatas2
'OvalShape14.Location = New Point(Val(TextBox19.Text),
Val(TextBox22.Text))
'Timer10.Enabled = True
End Sub
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    On Error GoTo iki
    soket2.SendData(TextBox7.Text)
Exit Sub
iki:    Label19.Text = "Disc"
    Button2.Enabled = False
    soket2.Close()
    soket2.Listen()
End Sub

'=====
'=====
Private Sub soket3_CloseEvent(ByVal sender As Object, ByVal e As
System.EventArgs) Handles soket3.CloseEvent
    Label10.Text = "Disc"
    soket3.Close()
    soket3.Listen()
    Button3.Enabled = False
End Sub
Private Sub soket3_ConnectionRequest(ByVal sender As Object, ByVal e As
AxMSWinsockLib.DMSWinsockControlEvents_ConnectionRequestEvent) Handles
soket3.ConnectionRequest
    soket3.Close()
    If soket3.RemoteHostIP = TextBox9.Text Then
        soket3.Accept(e.requestID)
    End If
End Sub

```

```

        Label10.Text = "OK"
        Button3.Enabled = True
    Else
        soket3.Close()
        soket3.Listen()
        Button3.Enabled = False
    End If
End Sub
Private Sub soket3_DataArrival(ByVal sender As Object, ByVal e As
AxMSWinsockLib.DMSWinsockControlEvents_DataArrivalEvent) Handles
soket3.DataArrival
    Dim sData As Byte
    Dim terima As String
    Dim jmlBuffer As Integer
    terima = ""
    jmlBuffer = e.bytesTotal

    While jmlBuffer > 0
        soket3.GetData(sData)
        terima = terima & Convert.ToChar(sData)
        jmlBuffer = jmlBuffer - 1
    End While

    TextBox10.Text = terima
End Sub
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
    On Error GoTo iki
    soket3.SendData(TextBox11.Text)
Exit Sub
iki:
    Label10.Text = "Disc"
    Button3.Enabled = False
    soket3.Close()
    soket3.Listen()
End Sub

'=====
=====
Private Sub soket4_CloseEvent(ByVal sender As Object, ByVal e As
System.EventArgs) Handles soket4.CloseEvent
    Label11.Text = "Disc"
    soket4.Close()
    soket4.Listen()
    Button4.Enabled = False
End Sub
Private Sub soket4_ConnectionRequest(ByVal sender As Object, ByVal e As
AxMSWinsockLib.DMSWinsockControlEvents_ConnectionRequestEvent) Handles
soket4.ConnectionRequest
    soket4.Close()
    If soket4.RemoteHostIP = TextBox12.Text Then
        soket4.Accept(e.requestID)
        Label11.Text = "OK"
        Button4.Enabled = True
    Else
        soket4.Close()
        soket4.Listen()
        Button4.Enabled = False
    End If
End Sub

```

```

        End If
    End Sub
    Private Sub socket4_DataArrival(ByVal sender As Object, ByVal e As
AxMSWinsockLib.DMSWinsockControlEvents_DataArrivalEvent) Handles
socket4.DataArrival
        Dim sData As Byte
        Dim terima As String
        Dim jmlBuffer As Integer
        terima = ""
        jmlBuffer = e.bytesTotal

        While jmlBuffer > 0
            socket4.GetData(sData)
            terima = terima & Convert.ToChar(sData)
            jmlBuffer = jmlBuffer - 1
        End While

        TextBox13.Text = terima
        Try
            Dim data As String() = terima.Split(";")
            'TextBox21.Text = (-data(0)) + 63 + 480 - 30
            'TextBox20.Text = (-data(1)) + 63 + 720 - 30
            TextBox21.Text = data(0) + 33
            TextBox20.Text = data(1) + 33
            R2X = data(0)
            R2Y = data(1)
            bolaatas2 = data(4)
            boladepan2 = data(5)
            kirimrobot2 = data(2) + "<" + code2b + ">" + data(3) + "|"
            kirimrobo1 = data(0) + "{" + data(1) + "}" + data(2) + "<" +
code2b + ">" + data(3) + "|"
        Catch ex As Exception

        End Try
        'OvalShape1.Location = New Point(Val(TextBox20.Text),
Val(TextBox21.Text))
        ' Timer10.Enabled = True
    End Sub
    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
        On Error GoTo iki
        socket4.SendData(TextBox14.Text)
    Exit Sub
iki:
    Label11.Text = "Disc"
    Button4.Enabled = False
    socket4.Close()
    socket4.Listen()
    End Sub
'=====Program Sebagai
Client=====
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        If Button1.Text = "Sambung" Then
            TimerAuto.Enabled = True
            Button1.Text = "Menghubungkan..."
        Else
            socket.Close()

```



```

        TimerAuto.Enabled = False
        Button1.Text = "Sambung"
    End If
End Sub
Private Sub soket_CloseEvent(ByVal sender As Object, ByVal e As
System.EventArgs) Handles soket.CloseEvent
    soket.Close()
    Button1.Text = "Menghubungkan..."
    TimerAuto.Enabled = True
End Sub
Private Sub soket_ConnectEvent(ByVal sender As Object, ByVal e As
System.EventArgs) Handles soket.ConnectEvent
    TimerAuto.Enabled = False
    Button1.Text = "Putuskan"
End Sub
Private Sub TimerAuto_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TimerAuto.Tick
    soket.Close()
    soket.Connect(TextBox1.Text, TextBox2.Text)
End Sub
Private Sub soket_DataArrival(ByVal sender As Object, ByVal e As
AxMSWinsockLib.DMSWinsockControlEvents_DataArrivalEvent) Handles
soket.DataArrival
    Dim sData As Byte
    Dim terima As String
    Dim jmlBuffer As Integer
    terima = ""
    jmlBuffer = e.bytesTotal

    While jmlBuffer > 0
        soket.GetData(sData)
        terima = terima & Convert.ToChar(sData)
        jmlBuffer = jmlBuffer - 1
    End While

    TextBox4.Text = terima
    If CheckBox1.Checked = False And CheckBox2.Checked = False Then
        TextBox7.Text = terima
        TextBox11.Text = terima
        TextBox14.Text = terima + "*" + TextBox28.Text + "{" +
TextBox29.Text + "}"
    End If
End Sub

'=====
=====
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
    soket2.Close()
    soket3.Close()
    soket4.Close()
    soket2.LocalPort = TextBox6.Text
    soket3.LocalPort = TextBox15.Text
    soket4.LocalPort = TextBox16.Text
    soket2.Listen()
    soket3.Listen()
    'soket4.Listen()

```

```

Label19.Text = "Disc"
Label10.Text = "Disc"
Label11.Text = "Disc"
Button2.Enabled = False
Button3.Enabled = False
Button4.Enabled = False
End Sub
Private Sub TimerKirim_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TimerKirim.Tick
    On Error GoTo error
    If CheckBox1.Checked = False And CheckBox2.Checked = False Then
        If Button2.Enabled = True Then
            soket2.SendData(TextBox7.Text)
        End If
        If Button3.Enabled = True Then
            soket3.SendData(TextBox11.Text)
        End If
        If Button4.Enabled = True Then
            soket4.SendData(TextBox14.Text)
        End If
    End If
Exit Sub
error: soket2.Close()
soket3.Close()
soket4.Close()
soket2.LocalPort = TextBox6.Text
soket3.LocalPort = TextBox15.Text
soket4.LocalPort = TextBox16.Text
soket2.Listen()
soket3.Listen()
soket4.Listen()
End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    soket2.Close()
    soket3.Close()
    soket4.Close()
    soket2.LocalPort = TextBox6.Text
    soket3.LocalPort = TextBox15.Text
    soket4.LocalPort = TextBox16.Text
    soket2.Listen()
    soket3.Listen()
    soket4.Listen()
    Label19.Text = "Disc"
    Label10.Text = "Disc"
    Label11.Text = "Disc"
    Button2.Enabled = False
    Button3.Enabled = False
    Button4.Enabled = False

End Sub
Private Sub Button11_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button11.Click
    TextBox33.Text = "K"

```

```
End Sub
Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    TextBox33.Text = "j"

End Sub
Private Sub Button10_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button10.Click
    TextBox33.Text = "p"

End Sub
Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button7.Click
    TextBox32.Text = "k"

End Sub
Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button8.Click
    TextBox32.Text = "w"

End Sub
Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button6.Click
    TextBox32.Text = "p"

End Sub
Private Sub CheckBox1_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CheckBox1.CheckedChanged
    If CheckBox1.Checked = True Then
        If Label11.Text = "OK" Then
            Button6.Enabled = True
            Button7.Enabled = True
            Button8.Enabled = True
            Button12.Enabled = True
            Button18.Enabled = True
            Button19.Enabled = True
            Button37.Enabled = True
            Button38.Enabled = True
            Button39.Enabled = True
            Button40.Enabled = True
            Button41.Enabled = True
            Button42.Enabled = True
            Button47.Enabled = True
            Button48.Enabled = True
            Button53.Enabled = True
            Button54.Enabled = True
            Button55.Enabled = True
            Button56.Enabled = True
            Button63.Enabled = True
            Timer1.Enabled = False
            Timer2.Enabled = False
            Timer3.Enabled = True
            Timer4.Enabled = False
            Timer6.Enabled = True
        End If
    Else
        Button6.Enabled = False
    End If
End Sub
```

```
Button7.Enabled = False
Button8.Enabled = False
Button12.Enabled = False
Button18.Enabled = False
Button19.Enabled = False
Button37.Enabled = False
Button38.Enabled = False
Button39.Enabled = False
Button40.Enabled = False
Button41.Enabled = False
Button42.Enabled = False
Button47.Enabled = False
Button48.Enabled = False
Button53.Enabled = False
Button54.Enabled = False
Button55.Enabled = False
Button56.Enabled = False
Button63.Enabled = False
Timer1.Enabled = True
Timer3.Enabled = False
Timer6.Enabled = False

End If
End Sub
Private Sub CheckBox2_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CheckBox2.CheckedChanged
If CheckBox2.Checked = True Then
If Label19.Text = "OK" Then
Button9.Enabled = True
Button10.Enabled = True
Button11.Enabled = True
Button13.Enabled = True
Button14.Enabled = True
Button15.Enabled = True
Button17.Enabled = True
Button32.Enabled = True
Button33.Enabled = True
Button34.Enabled = True
Button35.Enabled = True
Button36.Enabled = True
Button45.Enabled = True
Button46.Enabled = True
Button49.Enabled = True
Button50.Enabled = True
Button51.Enabled = True
Button52.Enabled = True
Button62.Enabled = True
Timer1.Enabled = False
Timer2.Enabled = True
Timer3.Enabled = False
Timer4.Enabled = False
Timer5.Enabled = True

End If
Else
Button9.Enabled = False
Button10.Enabled = False
```

```
Button11.Enabled = False
Button13.Enabled = False
Button14.Enabled = False
Button15.Enabled = False
Button19.Enabled = False
Button32.Enabled = False
Button17.Enabled = False
Button36.Enabled = False
Button33.Enabled = False
Button34.Enabled = False
Button35.Enabled = False
Button37.Enabled = False
Button38.Enabled = False
Button39.Enabled = False
Button40.Enabled = False
Button41.Enabled = False
Button42.Enabled = False
Button45.Enabled = False
Button46.Enabled = False
Button49.Enabled = False
Button50.Enabled = False
Button51.Enabled = False
Button52.Enabled = False
Button62.Enabled = False
Timer1.Enabled = True
Timer2.Enabled = False
Timer5.Enabled = False
End If
End Sub
Private Sub Button12_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button12.Click
    TextBox32.Text = "S"

End Sub
Private Sub Button13_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button13.Click
    TextBox33.Text = "S"

End Sub
Private Sub Button9_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button9.Click
    TextBox33.Text = "W"

End Sub
Private Sub Button14_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button14.Click
    TextBox33.Text = "X"

End Sub
Private Sub CheckBox2_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles CheckBox2.KeyPress
    If CheckBox2.Checked = True Then
        TextBox33.Text = e.KeyChar

End If
End Sub
```

```

Private Sub CheckBox2_KeyUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles CheckBox2.KeyUp
    If CheckBox2.Checked = True Then
        TextBox33.Text = "S"

    End If
End Sub

Private Sub CheckBox1_KeyUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles CheckBox1.KeyUp
    If CheckBox1.Checked = True Then
        TextBox32.Text = "S"

    End If
End Sub
Private Sub CheckBox1_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles CheckBox1.KeyPress
    If CheckBox1.Checked = True Then
        TextBox32.Text = e.KeyChar

    End If
End Sub
Private Sub CheckBox3_KeyUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles CheckBox3.KeyUp
    If CheckBox3.Checked = True Then
        TextBox32.Text = "S"

        TextBox33.Text = "S"

    End If
End Sub
Private Sub CheckBox3_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles CheckBox3.KeyPress
    If CheckBox3.Checked = True Then
        TextBox32.Text = e.KeyChar

        TextBox33.Text = e.KeyChar

    End If
End Sub

Private Sub Panel1_Paint(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles Panel1.Paint
'Dim ptLocationOnForm As New System.Drawing.Point
'Dim ptLocationOnScreen As New System.Drawing.Point
'ptLocationOnForm = Me.PointToClient(Cursor.Position)
'ptLocationOnScreen = Me.PointToScreen(Cursor.Position)
'TextBox27.Text = ptLocationOnForm.X.ToString - 601
'TextBox26.Text = ptLocationOnForm.Y.ToString - 54
'XKoordinat = Cursor.Position.X - 601
'YKoordinat = Cursor.Position.Y - 54
'TextBox27.Text = XKoordinat
'TextBox26.Text = YKoordinat
End Sub
Private Sub RectangleShape4_MouseClick(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.MouseEventArgs)

```

```
'End Sub

Private Sub RectangleShape4_MouseClick_1(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.MouseEventArgs) Handles
RectangleShape4.MouseClick
    XMouse = e.X
    YMouse = e.Y
End Sub
Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick
    TextBox27.Text = XMouse - 48
    TextBox26.Text = YMouse - 48
    X1 = (720 - (TextBox27.Text))
    Y1 = (480 - (TextBox26.Text))
    'TextBox25.Text = TextBox27.Text - TextBox22.Text
    'TextBox24.Text = TextBox26.Text - TextBox19.Text
    TextBox25.Text = TextBox27.Text
    TextBox24.Text = TextBox26.Text
    'TextBox31.Text = 0
    'TextBox30.Text = 0
End Sub

Private Sub Timer3_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer3.Tick
    TextBox31.Text = XMouse - 48
    TextBox30.Text = YMouse - 48
    X2 = (720 - (TextBox31.Text))
    Y2 = (480 - (TextBox30.Text))
    TextBox29.Text = TextBox31.Text
    TextBox28.Text = TextBox30.Text
    'TextBox27.Text = 0) * 100 / 6
    'TextBox26.Text = 0
End Sub

Private Sub CheckBox3_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CheckBox3.CheckedChanged
    If CheckBox3.Checked = True Then
        If Label9.Text = "OK" Then
            Button16.Enabled = True
            Button21.Enabled = True
            Button22.Enabled = True
            Button24.Enabled = True
            Button23.Enabled = True
            Button25.Enabled = True
            Button26.Enabled = True
            Button27.Enabled = True
            Button28.Enabled = True
            Button29.Enabled = True
            Button30.Enabled = True
            Button31.Enabled = True
            Button43.Enabled = True
            Button44.Enabled = True
            Button57.Enabled = True
            Button58.Enabled = True
            Button59.Enabled = True
            Button60.Enabled = True
            Button61.Enabled = True
        End If
    End If
End Sub
```

```
        Timer1.Enabled = False
        Timer5.Enabled = True
        Timer6.Enabled = True

    End If
Else
    Button16.Enabled = False
    Button21.Enabled = False
    Button22.Enabled = False
    Button24.Enabled = False
    Button23.Enabled = False
    Button25.Enabled = False
    Button26.Enabled = False
    Button27.Enabled = False
    Button28.Enabled = False
    Button29.Enabled = False
    Button30.Enabled = False
    Button31.Enabled = False
    Button43.Enabled = False
    Button44.Enabled = False
    Button57.Enabled = False
    Button58.Enabled = False
    Button59.Enabled = False
    Button60.Enabled = False
    Button61.Enabled = False
    Timer1.Enabled = True
    Timer5.Enabled = False
    Timer6.Enabled = False
End If
End Sub

Private Sub Button18_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button18.Click
    TextBox32.Text = "s"
    Timer10.Enabled = True
End Sub

Private Sub Button19_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button19.Click
    TextBox32.Text = "X"
End Sub

Private Sub Button15_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button15.Click
    TextBox33.Text = "s"
    Timer7.Enabled = True
    Timer8.Enabled = True
    stopwatch.Start()
    Timer10.Enabled = True
End Sub

Private Sub Timer4_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer4.Tick
    TextBox25.Text = 0
    TextBox24.Text = 0
End Sub
```



```
Private Sub Button23_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button23.Click

    TextBox33.Text = "s"

    TextBox32.Text = "s"
    Timer10.Enabled = True
End Sub

Private Sub Button24_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button24.Click
    TextBox33.Text = "w"

    TextBox32.Text = "w"

End Sub

Private Sub Button27_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button27.Click
    TextBox33.Text = "Q"

    TextBox32.Text = "Q"

End Sub

Private Sub Button25_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button25.Click
    TextBox33.Text = "c"

    TextBox32.Text = "c"

End Sub

Private Sub Button26_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button26.Click
    TextBox33.Text = "K"

    TextBox32.Text = "K"

End Sub

Private Sub Button22_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button22.Click
    TextBox33.Text = "S"

    TextBox32.Text = "S"

End Sub

Private Sub Button29_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button29.Click
    TextBox33.Text = "B"

    TextBox32.Text = "B"

End Sub
```

```
Private Sub Button31_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button31.Click
    TextBox33.Text = "g"

    TextBox32.Text = "g"

End Sub

Private Sub Button21_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button21.Click
    TextBox33.Text = "Z"

    TextBox32.Text = "Z"

End Sub

Private Sub Button28_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button28.Click
    TextBox33.Text = "X"

    TextBox32.Text = "X"

End Sub

Private Sub Button16_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button16.Click
    TextBox33.Text = "p"

    TextBox32.Text = "k"

End Sub

Private Sub GroupBox5_Enter(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles GroupBox5.Enter

End Sub

Private Sub Button33_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button33.Click
    TextBox33.Text = "Q"

End Sub

Private Sub Button32_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button32.Click
    TextBox33.Text = "B"

End Sub

Private Sub Button34_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button34.Click
    TextBox33.Text = "Z"

End Sub
```

```
Private Sub Button17_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button17.Click
    TextBox33.Text = "c"

End Sub

Private Sub Button36_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button36.Click
    TextBox33.Text = "g"

End Sub

Private Sub Button39_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button39.Click
    TextBox32.Text = "Q"

End Sub

Private Sub Button38_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button38.Click
    TextBox32.Text = "B"

End Sub

Private Sub Button40_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button40.Click
    TextBox32.Text = "Z"

End Sub

Private Sub Button42_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button42.Click
    TextBox32.Text = "c"

End Sub

Private Sub Button41_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button41.Click
    TextBox32.Text = "g"

End Sub

Private Sub Button30_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button30.Click
    TextBox33.Text = "b"

    TextBox32.Text = "b"

End Sub

Private Sub Button35_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button35.Click
    TextBox33.Text = "b"

End Sub
```

```
Private Sub Button37_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button37.Click
    TextBox32.Text = "b"

End Sub

Private Sub Button44_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button44.Click
    TextBox33.Text = "R"

    TextBox32.Text = "R"

End Sub

Private Sub Button43_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button43.Click
    TextBox33.Text = "T"

    TextBox32.Text = "T"

End Sub

Private Sub Button46_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button46.Click
    TextBox33.Text = "R"

End Sub

Private Sub Button45_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button45.Click
    TextBox33.Text = "T"

End Sub

Private Sub Button48_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button48.Click
    TextBox32.Text = "R"

End Sub

Private Sub Button47_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button47.Click
    TextBox32.Text = "T"

End Sub

Private Sub Button56_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button56.Click
    TextBox32.Text = "v"

End Sub

Private Sub Button55_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button55.Click
    TextBox32.Text = "v"

End Sub
```

```
Private Sub Button53_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button53.Click
    TextBox32.Text = "y"

End Sub

Private Sub Button54_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button54.Click
    TextBox32.Text = "Y"

End Sub

Private Sub Button52_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button52.Click
    TextBox33.Text = "v"

End Sub

Private Sub Button51_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button51.Click
    TextBox33.Text = "V"

End Sub

Private Sub Button50_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button50.Click
    TextBox33.Text = "y"

End Sub

Private Sub Button49_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button49.Click
    TextBox33.Text = "Y"

End Sub

Private Sub Button60_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button60.Click
    TextBox33.Text = "v"

    TextBox32.Text = "v"

End Sub

Private Sub Button59_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button59.Click
    TextBox33.Text = "V"

    TextBox32.Text = "V"

End Sub

Private Sub Button57_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button57.Click
    TextBox33.Text = "y"
```

```
        TextBox32.Text = "y"

    End Sub

    Private Sub Button58_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button58.Click
        TextBox33.Text = "Y"

        TextBox32.Text = "Y"

    End Sub

    Private Sub Button62_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button62.Click
        If Button62.Text = "Axis" Then
            Timer5.Enabled = False
            Timer9.Enabled = True
            Button62.Text = "False"
        Else
            Timer9.Enabled = False
            Timer5.Enabled = True
            Button62.Text = "Axis"
        End If
    End Sub

    Private Sub Button63_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button63.Click
        If Button63.Text = "Axis" Then
            Timer6.Enabled = False
            Timer11.Enabled = True
            Button63.Text = "False"
        Else
            Timer11.Enabled = False
            Timer6.Enabled = True
            Button63.Text = "Axis"
        End If
    End Sub

    Private Sub Button61_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button61.Click
        Timer5.Enabled = False
        Timer6.Enabled = False
        TextBox7.Text = "e" + "*" + TextBox24.Text + "{" + TextBox25.Text +
"}" + kirimrobot1
        TextBox14.Text = "e" + "*" + TextBox28.Text + "{" + TextBox29.Text
+ "}" + kirimrobot2
        soket4.SendData(TextBox14.Text)
        soket2.SendData(TextBox7.Text)
    End Sub

    Private Sub Timer5_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer5.Tick
        TextBox7.Text = TextBox33.Text + "*" + kirimrobot1
        soket2.SendData(TextBox7.Text)

    End Sub
```

```
Private Sub Timer6_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer6.Tick
    TextBox14.Text = TextBox32.Text + "*" + kirimrobot2
    socket4.SendData(TextBox14.Text)

End Sub

Private Sub Timer7_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer7.Tick
    Dim elapsed As TimeSpan = stopwatch.Elapsed
    Label17.Text = String.Format("{0:00}:{1:00}:{2:00}:{3:00}",
Math.Floor(elapsed.TotalHours), elapsed.Minutes, elapsed.Seconds,
elapsed.Milliseconds)
End Sub
Private Sub Button64_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button64.Click
    Timer7.Enabled = True
    Timer8.Enabled = True
    Timer10.Enabled = True
    stopwatch.Start()
End Sub

Private Sub Button65_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button65.Click
    Timer7.Stop()
    Timer8.Stop()
    stopwatch.Stop()
End Sub

Private Sub Button66_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button66.Click
    Timer7.Stop()
    Timer8.Stop()
    stopwatch.Reset()
    Label17.Text = "00:00:00:00"
End Sub

Private Sub PosisiRobotDataGridView_CellContentClick(ByVal sender As
System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs)
Handles PosisiRobotDataGridView.CellContentClick
    If PosisiRobotDataGridView.Enabled = True Then
        OvalShape14.Location = New Point(Val((TextBox35.Text / 100 * 8)
+ 33), Val((TextBox34.Text / 100 * 8) + 33))
        OvalShape1.Location = New Point(Val((TextBox36.Text / 100 * 8)
+ 33), Val((TextBox37.Text / 100 * 8) + 33))
        Timer10.Enabled = False

    Else
        Timer10.Enabled = True
    End If

End Sub

Private Sub Timer8_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer8.Tick
    PosisiRobotBindingSource.AddNew()
```

```
TimerTextBox.Text = Label17.Text
TextBox34.Text = R1X * 100 / 8
TextBox35.Text = R1Y * 100 / 8

TextBox37.Text = R2X * 100 / 8
TextBox36.Text = R2Y * 100 / 8
PosisiRobotBindingSource.EndEdit()

PosisiRobotTableAdapter.Update(DatabasePosisiRobotDataSet.PosisiRobot)
End Sub

Private Sub Timer9_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer9.Tick
    TextBox7.Text = "e" + "*" + TextBox24.Text + "{" + TextBox25.Text +
"}" + kirimrobot1
    soket2.SendData(TextBox7.Text)
End Sub

Private Sub Label6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Label6.Click

End Sub

Private Sub Timer10_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer10.Tick
    OvalShape1.Location = New Point(Val(TextBox20.Text),
Val(TextBox21.Text))
    OvalShape14.Location = New Point(Val(TextBox19.Text),
Val(TextBox22.Text))
End Sub

Private Sub Timer11_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer11.Tick
    TextBox14.Text = "e" + "*" + TextBox28.Text + "{" + TextBox29.Text
+ "}" + kirimrobot2
    soket4.SendData(TextBox14.Text)
End Sub

Private Sub TextBox23_TextChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles TextBox23.TextChanged

End Sub

Private Sub TextBox18_TextChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles TextBox18.TextChanged

End Sub

End Class
```


Lampiran 10. Program RTU**a. Program Teensy 4.0**

```

#include "Wire.h"
#include "Odometry.h"
#include "Caribola.h"

Odometry odo;
Caribola cari;

Servo servoku;
Servo servomu;

double ex, ey;
char data2[13] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
char code;
int kalib, klbx, klbx2, klbx3, klbx4, klby, klby2, klby3, klby4,
minus2x, minus2y;
double x_terima, y_terima, t_x, t_y, sudutPos, sudutPos2, sudutHadap;
const int ping_pin = 21; // ping
const int ping_pin2 = 20;
int rotaryx1, rotaryx2, rotaryx3, rotaryx4, rotaryy1, rotaryy2,
rotaryy3, rotaryy4;
int minusx, minusy, minusk, minuss;
long durasi, durasi2, inches, inches2, cm, cm2;
int keputusan, giring, langkah;
int codek, code3 , cobacode3;
char code2;
int Sudut1, nY, nX, kX, kY;
int S_udut2, S_udut3;
int hitung, hitung2, transisi;

#define encoder0PinA 8
#define encoder0PinB 9
#define encoder1PinA 10
#define encoder1PinB 11

#define pwm1 0
#define pwm2 1
#define pwm3 2
#define pwm4 3
#define pwm5 4
#define pwm6 5

#define giring1 12
#define giring2 13
#define giring3 23
#define giring4 22

void setup() {
  Serial.begin (115200);
  Wire.begin();

  pinMode(pwm1, OUTPUT);

```

```
pinMode(pwm2, OUTPUT);
pinMode(pwm3, OUTPUT);
pinMode(pwm4, OUTPUT);
pinMode(pwm5, OUTPUT);
pinMode(pwm6, OUTPUT);
pinMode(12, OUTPUT);
pinMode(13, OUTPUT);
pinMode(22, OUTPUT);
pinMode(23, OUTPUT);

pinMode(encoder0PinA, INPUT_PULLUP);
pinMode(encoder0PinB, INPUT_PULLUP);
pinMode(encoder1PinA, INPUT_PULLUP);
pinMode(encoder1PinB, INPUT_PULLUP);
}

void loop() {
  mulai();
}

void gerakan() {
  if (gerak == 1) {
    save = 0;
    save1 = 0;
    cariSkor(lambung, banter);
  }
  else if (gerak == 2) {
    save = 0;
    save1 = 0;
    if (strategy == 0 || strategy == 1) {
      kordinat(700, 10700, 50, anglek); //CORNER KIRI 2
    }
    else {
      kordinat(3000, 2000, 0, angleH); //CORNER KIRI 2
    }
  }
  else if (gerak == 3) {
    save = 0;
    save1 = 0;
    if (code2 == 'p') {
      // halang3_kanan();
      cariSkor(lambung, banter);
    }
    else if (code2 == 'b') {
      motor(0,0,0);
    }
  }

  // servoku.write(gaknendang);
  // kordinat(3000, 1500, 45, anglek); //CORNER MUSUH 3
}
else if (gerak == 4) {
  save = 0;
  save1 = 0;
  if (strategy == 0 || strategy == 1) {
    kordinat(3000, 3500, 0, angleH); //DROP BALL 4
```

```
    }
    else {
        kordinat(3300, 2000, 0, angleH); //DROP BALL 4
    }
}
else if (gerak == 5) {
    save = 0;
    save1 = 0;
    if (strategy == 0 || strategy == 1) {
        kordinat(4500, 3500, 0, angleH); //DB KANAN 5
    }
    else {
        kordinat(3000, 1500, 0, angleH); //DB KANAN 5
    }
}
else if (gerak == 6) {
    save = 0;
    save1 = 0;
    if (strategy == 0 || strategy == 1) {
        kordinat(1500, 3500, 0, angleH); //DB KIRI 6
    }
    else {
        kordinat(3000, 1500, 0, angleH); //DB KIRI 6
    }
}
else if (gerak == 7) {
    save = 0;
    save1 = 0;
    kordinat(3000, 4000, 0, anglek); //GOLL KICK 7
}
else if (gerak == 8) {
    save = 0;
    save1 = 0;
    kordinat(1500, 4500, 0, angleH);
    // kordinat(4500, 1300, 0, angleH); //GK KANAN 8
}
else if (gerak == 9) {
    save = 0;
    save1 = 0;
    kordinat(1500, 1300, 0, angleH); //GK KIRI 9
}
else if (gerak == 10) {
    save = 0;
    save1 = 0;
    kordinat(3000, 3500, 0, angleH); //KICK OFF 10
}
else if (gerak == 11) {
    n = 0;
    save1 = 0;
    nilai_ping = 1;
    { if (save < 1) {
        a = 1;
        xhinder = x;
        yhinder = y;
    }
```

```
        kondisi2 = batas_S;
        suduthindar = angle90;
        suduthindar1 = angleH;
        save++;
    }
    else if (save >= 1) {
        a = a;
        save = 1;
        xhindar = xhindar;
        yhindar = yhindar;
        kondisi2 = kondisi2;
        suduthindar = suduthindar;
        suduthindar1 = suduthindar1;
        halang1_kanan();
    }
}
giringo(175, 175);
if (code2 == 'p') {
    halang3_kanan();
}
else if (code2 == 'b') {
    motor(0,0,0);
}
}
else if (gerak == 12) {
    save = 0;
    save1 = 0;
    if (strategy == 0 || strategy == 1) {
        kordinat(3000, 3000, 0, angleH); //GOAL 12
    }
    else {
        kordinat(3000, 2500, 0, angleH); //GOAL 12
    }
}
else if (gerak == 13) {
    save = 0;
    save1 = 0;
    kordinat(x, 2000, 0, angleH); //MUNDUR 13
}
else if (gerak == 14) {
    save = 0;
    save1 = 0;
    kordinat(2000, 1000, 0, angleH); //MUNDURR 14
}
else if (gerak == 15) {
    //-----HINDAR 15
    n = 0;
    save1 = 0;
    nilai_ping = 1;
    { if (save < 1) {
        a = 1;
        xhindar = x;
        yhindar = y;
        kondisi2 = batas_S;
```

```
        suduthindar = angle90;
        suduthindar1 = angleH;
        save++;
    }
    else if (save >= 1) {
        a = a;
        save = 1;
        xhindar = xhindar;
        yhindar = yhindar;
        kondisi2 = kondisi2;
        suduthindar = suduthindar;
        suduthindar1 = suduthindar1;
        halang1_kanan();
    }
}
{ if (kondisi2 == 1) {
    halang3_kanan();
}
  else if (kondisi2 == 2) {
    halang3_kiri();
}
  else if (kondisi2 == 31) {
    halang2_kanan();
}
  else if (kondisi2 == 32) {
    halang1_kanan();
}
  else if (kondisi2 == 41) {
    halang2_kiri();
}
  else if (kondisi2 == 42) {
    halang1_kiri();
}
}
}
else if (gerak == 16) {
    //-----BATAS DEPAN KIRI 16
    a = 1;
    save = 0;
    save1 = 0;
    n = 0;
    p_giringan = 1;
    kordinat2(1500, 7500, 0, angleH, 60, 100);
    if (sel <= 70 ) nilai_ping = 1;
    else nilai_ping = 0;
}
else if (gerak == 17) {
    //-----BATAS DEPAN KANAN 17
    a = 1;
    save = 0;
    save1 = 0;
    n = 0;
    p_giringan = 1;
    kordinat2(4500, 7500, 0, angleH, 60, 100);
```

```
    if (sel <= 70 ) nilai_ping = 1;
    else nilai_ping = 0;
}
else if (gerak == 18) {
//-----BATAS DEPAN TENGAH 18
a = 1;
save = 0;
save1 = 0;
n = 0;
p_giringan = 1;
koordinat2(X, 7500, 0, angleH, 60, 100);
if (sel <= 20 ) nilai_ping = 1;
else nilai_ping = 0;
}
else if (gerak == 19) {
//-----DINDING KIRI 19
save = 0;
n = 0;
p_giringan = 1;
{
    if (save1 < 1) {
        a = 1;
        xhindar = x;
        yhindar = y;
        suduthindar = angle90;
        suduthindar1 = angleH;
        save1++;
    }
    else if (save1 >= 1) {
        a = a;
        save1 = 1;
        xhindar = xhindar;
        yhindar = yhindar;
        suduthindar = suduthindar;
        suduthindar1 = suduthindar1;
    }
}
//    nilai_ping = 1;
halang3_kanan();
}
else if (gerak == 20) {
//-----DINDING KANAN 20
n = 0;
p_giringan = 1;
{
    if (save1 < 1) {
        a = 1;
        xhindar = x;
        yhindar = y;
        suduthindar = angle90;
        suduthindar1 = angleH;
        save1++;
    }
    else if (save1 >= 1) {
```

```
        a = a;
        save1 = 1;
        xhindar = xhindar;
        yhindar = yhindar;
        suduthindar = suduthindar;
    }
}
// nilai_ping = 1;
halang3_kiri();
}
else if (gerak == 21) {
//-----MUNDUR B 21
a = 1;
kordinat(3000, 2500, angleH, angleH);
if (sel <= 100 ) {
    nilai_ping = 1;
}
}
else if (gerak == 22) {
//-----MUNDUR A 22
a = 1;
kordinat(3000, 4500, angleH, angleH);
if (sel <= 100 ) {
    nilai_ping = 1;
}
}
else if (gerak == 23) {
//----- 23
a = 1;
kordinat(3000, 1000, angleH, angleH);
if (sel <= 100 ) {
    nilai_ping = 1;
}
}
else if (gerak == 24) {
//----- 24
a = 1;
kordinat(x, 3000, 0, angleH);
if (sel <= 100 ) {
    nilai_ping = 1;
}
}
else if (gerak == 25) {
//----- 25
a = 1;
kordinat(y, 1500, 0, angleH);
if (sel <= 100 ) {
    nilai_ping = 1;
}
}
else if (gerak == 26) {
//-----UMPAN 26
n = 0;
save1 = 0;
```

```
nilai_ping = 1;
if (save < 1) {
    a = 1;
    xhindar = x;
    yhindar = y;
    kondisi2 = batas_S;
    suduthindar = angle90;
    suduthindar1 = angleH;
    save++;
}
else if (save >= 1) {
    a = a;
    save = 1;
    xhindar = xhindar;
    yhindar = yhindar;
    kondisi2 = kondisi2;
    suduthindar = suduthindar;
    suduthindar1 = suduthindar1;
    umpan();
}
}
else if (gerak == 27) {
    if (ex >= 3000 && ey <= 1200) {
        kordinat(ex - 1500, ey + 700, Sudut2, angleH);
    }
    else if (ex < 3000 && ey <= 1200) {
        kordinat(ex + 1500, ey + 700, Sudut2, angleH);
    }
    else if (ex >= 3000 && ey > 1200) {
        kordinat(ex - 1500, ey - 700, Sudut2, angleH);
    }
    else if (ex < 3000 && ey > 1200) {
        kordinat(ex + 1500, ey - 700, Sudut2, angleH);
    }
}
else if (gerak == 28) {
    if (ex >= 3000 && ey <= 1200) {
        kordinat(ex - 2000, ey + 700, Sudut2, angleH);
    }
    else if (ex < 3000 && ey <= 1200) {
        kordinat(ex + 2000, ey + 700, Sudut2, angleH);
    }
    else if (ex >= 3000 && ey > 1200) {
        kordinat(ex - 2000, ey - 1500, Sudut2, angleH);
    }
    else if (ex < 3000 && ey > 1200) {
        kordinat(ex + 2000, ey - 1500, Sudut2, angleH);
    }
}
else if (gerak == 29) {

}
else {
    nilai_ping = 0;
}
```



```
pwmMax = 0;
a = 1;
save = 0;
save1 = 0;

}
}

void terima_umpan() {
{
if (ex < 3000 && ey < 4500) {
if (cm < 4 && cm2 < 4) {
sudut90 = sudut90 - 5;
}
else {
sudut90 = sudut90;
}
if (sudut90 < 0) sudut90 += 360;
sudutPos = sudut90;
}
else if (ex > 3000 && ey < 4500) {
if (cm < 4 && cm2 < 4) {
sudut270 = sudut270 + 5;
}
else {
sudut270 = sudut270;
}
if (sudut270 >= 360) sudut270 -= 360;
sudutPos = sudut270;
}
else if (ex < 3000 && ey > 4500) {
if (cm < 4 && cm2 < 4) {
sudut90 = sudut90 + 5;
}
else {
sudut90 = sudut90;
}
if (sudut90 >= 360) sudut90 -= 360;
sudutPos = sudut90;
}
else if (ex > 3000 && ey > 4500) {
if (cm < 4 && cm2 < 4) {
sudut270 = sudut270 - 5;
}
else {
sudut270 = sudut270;
}
if (sudut270 < 0) sudut270 += 360;
sudutPos = sudut270;
}
}
}
```

```
Sudut2 = (atan2((ex - x), (ey - y)) * 180 / PI) * -1; // sudut
menerima umpan
{
  if (sudutPos < 120 || sudutPos > 240) {
    sudutPos2 = angleH;
    sudutHadap = Sudut2;
  }
  else if (sudutPos > 120 || sudutPos < 240) {
    sudutPos2 = anglek;
    if (Sudut2 < 0) {
      sudutHadap = Sudut2 + 180;
    }
    else {
      sudutHadap = Sudut2 - 180;
    }
  }
}
t_x = sin(radians(sudutPos)) * 2000; // posisi robot terhadap kawan
t_y = cos(radians(sudutPos)) * 2000;
x_terima = ex + t_x;
y_terima = ey + t_y;

kordinat(x_terima, y_terima, sudutHadap, sudutPos2);
}

void xyhindar(int hindari) {
  xhindar_p = hindari * cos(radians(suduthindar));
  yhindar_p = hindari * sin(radians(suduthindar));
}

void hindar_kanan() {
  xyhindar(-300);
  if (a == 1) {
    giringo(175, 175);
    kordinat2(xhindar + xhindar_p, yhindar + yhindar_p, suduthindar1,
angleH, 60, 80);
    if (sel <= 70 ) {
      a++;
    }
  }
  else if (a == 2) {
    giringo(175, 175);
    kordinat2(xhindar + xhindar_p, yhindar + yhindar_p, 160, angleH,
30, 80);
    if (pidacuan >= pid_setpoint - 15 && pidacuan <= pid_setpoint +
15) a++;
  }
  else if (a == 3) {
    giringo(175, 175);
    kordinat2(xhindar + 1000, yhindar + yhindar_p, 90, anglek, 30,
70);
    if (sel <= 140) a++;
  }
  else if (a == 4) {
```

```
        giringo(175, 175);
        kordinat2(xhindar + 2000, yhindar, 0, angleH, 30, 70);
        if (sel <= 70 ) a++;
    }
    else if (a == 5) {
        giringo(175, 175);
    }
}

void hindar_kiri() {
    xyhindar(-300);
    if (a == 1) {
        giringo(175, 175);
        kordinat2(xhindar + xhindar_p, yhindar + yhindar_p, suduthindar1,
angleH, 60, 80);
        if (sel <= 70 ) {
            a++;
        }
    }
    else if (a == 2) {
        giringo(175, 175);
        kordinat2(xhindar + xhindar_p, yhindar, -160, angleH, 30, 80);
        if (pidacuan >= pid_setpoint - 15 && pidacuan <= pid_setpoint +
15) a++;
    }
    else if (a == 3) {
        giringo(175, 175);
        kordinat2(xhindar - 1000, yhindar, -90, anglek, 30, 70);
        if (sel <= 140) a++;
    }
    else if (a == 4) {
        giringo(175, 175);
        kordinat2(xhindar - 2000, yhindar, 0, angleH, 30, 70);
        if (sel <= 70 ) a++;
    }
    else if (a == 5) {
        giringo(175, 175);
    }
}

void umpan() {
    xyhindar(-300);
    if (a == 1) {
        codek = 2;
        nilai_ping = 0;
        p_giringan = 1;
        kordinat2(xhindar + xhindar_p, yhindar + yhindar_p, suduthindar1,
angleH, 60, 80);
        if (sel <= 70 ) a++;
    }
    else if (a == 2) {
        codek = 2;
        p_giringan = 1;
    }
}
```

```

    kordinat2(xhindar + xhindar_p, yhindar + yhindar_p, 0, anglek,
30, 80);
    if (pidacuan >= pid_setpoint - 5 && pidacuan <= pid_setpoint +
5) a++;
}
else if (a == 3) {
    if (code3 == 0) {
        giringo(175, 175);
        motor(0, 0, 0);
    }
    else if (code3 == 1) { // posisi robot B siap
        giringo(175, 175);
        cariKawan();
        if (Suduts2 >= pid_setpoint1 - 2 && Suduts2 <= pid_setpoint1 +
2) {
            codek = 3;
        }
    }
    else if (code3 == 2) { // tendang umpan
        codek = 3;
        cariKawan();
        hitung++;
        transisi = 1;
        if (hitung > 120) {
            if (Suduts2 >= pid_setpoint1 - 2 && Suduts2 <= pid_setpoint1
+ 2) {
                giringo(0, 0);
                servoku.write(pelan);
            }
            else if (Suduts2 < pid_setpoint1 - 2 || Suduts2 >
pid_setpoint1 + 2) {
                nwaktu = 0;
                servoku.write(gaknendang);
            }
        }
    }
    else{
        servoku.write(gaknendang);
    }
}
else {
    giring = 0;
    codek = 1;
    keputusan = 1;
}
}
}

void kom_millis() {
    kX = (x / 100) * 8; //komunikasi vb
    if (kX < 0) minusx = 0;
    else minusx = 1;

    kY = (y / 100) * 8;

```

```

    if (kY < 0) minusy = 0;
    else minusy = 1;

    rotaryx1 = abs(kX);
    if (rotaryx1 >= 255) rotaryx1 = 255;
    else if (rotaryx1 <= 0) rotaryx1 = 0;
    else rotaryx1 = rotaryx1;
    rotaryx2 = abs(kX) - 255;
    if (rotaryx2 >= 255) rotaryx2 = 255;
    else if (rotaryx2 <= 0) rotaryx2 = 0;
    else rotaryx2 = rotaryx2;
    rotaryx3 = abs(kX) - 510;
    if (rotaryx3 >= 255) rotaryx3 = 255;
    else if (rotaryx3 <= 0) rotaryx3 = 0;
    else rotaryx3 = rotaryx3;
    rotaryx4 = abs(kX) - 765;
    if (rotaryx4 >= 255) rotaryx4 = 255;
    else if (rotaryx4 <= 0) rotaryx4 = 0;
    else rotaryx4 = rotaryx4;

    rotaryy1 = abs(kY);
    if (rotaryy1 >= 255) rotaryy1 = 255;
    else if (rotaryy1 <= 0) rotaryy1 = 0;
    else rotaryy1 = rotaryy1;
    rotaryy2 = abs(kY) - 255;
    if (rotaryy2 >= 255) rotaryy2 = 255;
    else if (rotaryy2 <= 0) rotaryy2 = 0;
    else rotaryy2 = rotaryy2;
    rotaryy3 = abs(kY) - 510;
    if (rotaryy3 >= 255) rotaryy3 = 255;
    else if (rotaryy3 <= 0) rotaryy3 = 0;
    else rotaryy3 = rotaryy3;
    rotaryy4 = abs(kY) - 765;
    if (rotaryy4 >= 255) rotaryy4 = 255;
    else if (rotaryy4 <= 0) rotaryy4 = 0;
    else rotaryy4 = rotaryy4;
}

void mulai() {
    Wire.requestFrom(9, 25);
    fP = Wire.read(); fp = Wire.read(); fB = Wire.read();
    fb = Wire.read(); cP = Wire.read(); cp = Wire.read();
    cB = Wire.read(); cb = Wire.read(); button = Wire.read();
    code = Wire.read(); klbx = Wire.read(); klbx2 = Wire.read();
    klbx3 = Wire.read(); klbx4 = Wire.read(); klby = Wire.read();
    klby2 = Wire.read(); klby3 = Wire.read(); klby4 = Wire.read();
    minus2x = Wire.read(); minus2y = Wire.read(); sudutk1 = Wire.read();
    sudutk2 = Wire.read(); code2 = Wire.read(); code3 = Wire.read();

    data2[0] = rotaryx1; data2[1] = rotaryx2; data2[2] = rotaryx3;
    data2[3] = rotaryx4; data2[4] = rotaryy1; data2[5] = rotaryy2;
    data2[6] = rotaryy3; data2[7] = rotaryy4; data2[8] = minusx;
    data2[9] = minusy; data2[10] = codek; data2[11] = sudut1;
    data2[12] = sudut2;
}

```

```

Wire.beginTransmission(2);
Wire.write(data2, 13);
Wire.endTransmission();

awalan(); baca_ping();
kX = odo.posisi_X(); kY = odo.posisi_Y();
}

void awalan() {
  int nStart;
  { //-----kalibrasi-----
  ----
    if (code == 'e') odo.kalibposisi(ex, ey);
    else if (code == 'p') odo.kalibcmps();
  }

  { //-----kontrol manual-----
  ----
    if (code == 'S') motor(0, 0, 0);
    else if (code == 'l') motor(50, 50, 50);
    else if (code == 'j') motor(-50, -50, -50);
    else if (code == 'i') motor(100, -100, 0);
    else if (code == 'm') motor(-115, 115, 0);
    else if (code == 'u') motor(-63, -63, 125);
    else if (code == 'o') motor(63, 63, -125);
  }

  { //-----posisi persiapan-----
  ----
    if (code == 's') {
      if (code2 == 'p') Penyerang();
      else if (code2 == 'b') Bertahan();
      else motor(0, 0, 0);
    }
    if (code == 'Q') { //corner kanan
      odo.koordinat_ckanan();
    }
    else if (code == 'W') { //corner kiri
      odo.koordinat_ckiri();
    }
    else if (code == 'c') { //corner musuh
      odo.koordinat_cmusuh();
    }
    else if (code == 'N') { //drop ball
      odo.koordinat_dball();
    }
    else if (code == 'b') { //gol kick kanan
      odo.koordinat_gkkanan();
    }
    else if (code == 'B') { //gol kick kiri
      odo.koordinat_gkkiri();
    }
    else if (code == 'K') { //kickoff sendiri

```

```

        odo.koordinat_koff();
    }
    else if (code == 'k') { //kickoff musuh
        odo.koordinat_koffm();
    }
    else if (code == 'g') { //GK Musuh
        odo.koordinat_gkm();
    }
    else if (code == 'S') {
        motor(0, 0, 0);
        giringo(0, 0);
    }
}
}

void sudutBola() {
    if (cP == 0)P = fP * -1; else P = fP;
    if (cp == 0)p = fp * -1; else p = fp;
    if (cB == 0)B = fB * -1; else B = fB;
    if (cb == 0)b = fb * -1; else b = fb;
}

void kalibrasi_pos() {
    ex = (klbx + klbx2 + klbx3 + klbx4) * 100 / 8;
    if (minus2x == 0) ex = ex * -1;
    else ex = ex;

    ey = (klby + klby2 + klby3 + klby4) * 100 / 8;
    if (minus2y == 0) ey = ey * -1;
    else ey = ey;
}

void Bertahan() {
    if (transisi == 0) {
        if (code3 == 0) { // back up normal
            odo.bertahan_back1();
        }
        else if (code3 == 1) { // back up saat menggiring
            odo.bertahan_back2();
        }

        else if (code3 == 2) { // persiapan menerima umpan
            odo.posisiterima_umpan();
        }

        else if (code3 == 3) { // siap menerima umpan
            odo.terima_umpan
        }
    }
}

void Penyerang() { // -----
    -----
    if (B <= 15 && B >= -15 && b <= 35) {

```

```
    if ((cm <= 45 || cm2 <= 45) && code3 == 3) {
        odo.giring();
    }
    else if (cm <= 45 || cm2 <= 45) {
        odo.umpan();
    }
    else {
        odo.carigawang();
    }
}
else if ((B < 60 || B > -60) && b <= 70 && prox == 1) {
    cari.bola(B, b, P, p);
}
else {
    cari.bola(B, b, P, p);
}
}

void motor(int tri, int one, int two) {
{ if (one > 0) {
    analogWrite(pwm1, one);
    analogWrite(pwm2, 0);
}
else if (one < 0) {
    analogWrite(pwm1, 0);
    analogWrite(pwm2, abs(one));
}
else {
    analogWrite(pwm1, 0);
    analogWrite(pwm2, 0);
}
}
{ if (two > 0) {
    analogWrite(pwm3, two);
    analogWrite(pwm4, 0);
}
else if (two < 0) {
    analogWrite(pwm3, 0);
    analogWrite(pwm4, abs(two));
}
else {
    analogWrite(pwm3, 0);
    analogWrite(pwm4, 0);
}
}
}
{ if (tri > 0) {
    analogWrite(pwm5, tri);
    analogWrite(pwm6, 0);
}
else if (tri < 0) {
    analogWrite(pwm5, 0);
    analogWrite(pwm6, abs(tri));
}
else {
```



```
        analogWrite(pwm5, 0);
        analogWrite(pwm6, 0);
    }
}

void giringo(int pwm_2, int pwm_1) {
{ if (pwm_1 > 0) {
    analogWrite(giring3, 0);
    analogWrite(giring4, pwm_1);
}
else if (pwm_1 < 0) {
    analogWrite(giring3, abs(pwm_1));
    analogWrite(giring4, 0);
}
else if (pwm_1 == 0) {
    analogWrite(giring3, 0);
    analogWrite(giring4, 0);
}
}
{ if (pwm_2 > 0) {
    analogWrite(giring1, 0);
    analogWrite(giring2, pwm_2);
}
else if (pwm_2 < 0) {
    analogWrite(giring1, abs(pwm_2));
    analogWrite(giring2, 0);
}
else if (pwm_2 == 0) {
    analogWrite(giring1, 0);
    analogWrite(giring2, 0);
}
}
}

void baca_ping() {
    pinMode(ping_pin, OUTPUT);
    digitalWrite(ping_pin, LOW);
    pinMode(ping_pin2, OUTPUT);
    digitalWrite(ping_pin2, LOW);
    delay(2);
    digitalWrite(ping_pin, HIGH);
    digitalWrite(ping_pin2, HIGH);
    delay(5);
    digitalWrite(ping_pin, LOW);
    pinMode(ping_pin, INPUT);
    durasi = pulseIn(ping_pin, HIGH);
    digitalWrite(ping_pin2, LOW);
    pinMode(ping_pin2, INPUT);
    durasi2 = pulseIn(ping_pin2, HIGH);

    cm = microsecondsToCentimeters(durasi);
    cm2 = microsecondsToCentimeters2(durasi2);
}
```

```

long microsecondsToCentimeters(long microseconds) {
    return microseconds / 29 / 2;
}

long microsecondsToCentimeters2(long microseconds) {
    return microseconds / 29 / 2;
}

```

b. Program Arduino Nano1

```

#include <Wire.h>
int gerak;
char data[13] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
int rotaryx1, rotaryx2, rotaryx3, rotaryx4, rotaryy1, rotaryy2,
rotaryy3, rotaryy4;
int minusx, minusy;
int kX, kY, sudut, sudut1, sudut2;
int codes;

void setup(){
    Wire.begin(2);
    Wire.onReceive(receiveEvent);
    Serial.begin(115200);
}

void loop(){
    rotary();
    sudutt();
    String data2 = String (kX) + ";" + String (kY) + ";" + String
(sudut) + ";" + String (codes) + ";";
    Serial.println(data2);
    delay(100);
}

void receiveEvent(int howMany){
    rotaryx1 = Wire.read(); rotaryx2 = Wire.read();
    rotaryx3 = Wire.read(); rotaryx4 = Wire.read();
    rotaryy1 = Wire.read(); rotaryy2 = Wire.read();
    rotaryy3 = Wire.read(); rotaryy4 = Wire.read();
    minusx = Wire.read(); minusy = Wire.read();
    codes = Wire.read(); sudut1 = Wire.read();
    sudut2 = Wire.read();
}

void rotary(){
    kX = rotaryx1 + rotaryx2 + rotaryx3 + rotaryx4;
    if(minusx == 0) kX = kX * -1;
    else kX = kX;

    kY = rotaryy1 + rotaryy2 + rotaryy3 + rotaryy4;
    if(minusy == 0) kY = kY * -1;
}

```

```

    else kY = kY;
}

void sudutt() {
    sudut = sudut1 + sudut2;
}

```

c. Program Arduino Nano2

```

#include <Wire.h>
boolean done = false;
int button, button1, button2, button3, button4, B, o, b, L, l, p, O,
d, P, cP, cp, cB, cb = 0;
char code, code2, code3;
const int ping_pin = 6;
const int ping_pin2 = 7;
unsigned long previousMicros = 0, previousMillis = 0, previousKom =
0;
const long interval = 8;
long durasi, durasi2, inches, inches2, cm, cm2;
unsigned long currentMillis = millis();
int gerak;
int klbx, klbx2, klbx3, klbx4, klby, klby2, klby3, klby4, minus2x,
minus2y;
int kalia, klax, klax2, klax3, klax4, klay, klay2, klay3, klay4,
minas2x, minas2y, minusk;
long ex, ey, ax, ay, sudutk, sudutk1, sudutk2;

void setup() {
    Wire.begin(9);
    Wire.onRequest(requestEvent);
    Serial.begin(115200);
    pinMode(9, INPUT); pinMode(10, INPUT);
    pinMode(11, INPUT); pinMode(12, INPUT);
}

void loop() {
    // String data = String (100) + ";" + String (100) + ";" ;
    // Serial.println(data);
    Serial.flush();
    (Serial.available() == 0);
    if (Serial.available()) {
        if (Serial.read() == '(') {
            String str8 = Serial.readStringUntil(')');
            String str11 = Serial.readStringUntil('!');
            String str1 = Serial.readStringUntil('@');
            String str2 = Serial.readStringUntil('#');
            String str3 = Serial.readStringUntil('$');
            String str4 = Serial.readStringUntil('%');
            String str5 = Serial.readStringUntil('^');
            String str6 = Serial.readStringUntil('&');
            String str9 = Serial.readStringUntil('*');
            String str12 = Serial.readStringUntil('{');

```

```

String str13 = Serial.readStringUntil('}');
String str14 = Serial.readStringUntil('>');
String str15 = Serial.readStringUntil('<');
String str16 = Serial.readStringUntil('|');

P = str1.toInt(); p = str2.toInt(); L = str3.toInt();
l = str4.toInt(); 0 = str5.toInt(); o = str6.toInt();
B = str8.toInt(); b = str11.toInt(); ex = str12.toInt();
ey = str13.toInt(); sudutk= str15.toInt(); code = str9[0];
code2 = str14[0]; code3 = str16[0];
Serial.flush();

    if (P >= 0)cP = 1; else cP = 0;
    if (p >= 0)cp = 1; else cp = 0;
    if (B >= 0)cB = 1; else cB = 0;
    if (b >= 0)cb = 1; else cb = 0;

}
}
button1 = digitalRead(9); button2 = digitalRead(10);
button3 = digitalRead(11); button4 = digitalRead(12);
if (button1 == HIGH && button2 == LOW && button3 == LOW && button4
== LOW) {
    button = 1;
}
else if (button1 == LOW && button2 == HIGH && button3 == LOW &&
button4 == LOW) {
    button = 2;
}
else if (button1 == LOW && button2 == LOW && button3 == HIGH &&
button4 == LOW) {
    button = 3;
}
else if (button1 == LOW && button2 == LOW && button3 == LOW &&
button4 == HIGH) {
    button = 4;
}
else if (button1 == LOW && button2 == LOW && button3 == LOW &&
button4 == LOW) {
    button = 0;
}
data_klbpos();
}

void requestEvent() {
    Wire.write(abs (P)); Wire.write(abs (p));
    Wire.write(abs (B)); Wire.write(abs (b));
    Wire.write(abs (cP)); Wire.write(abs (cp));
    Wire.write(abs (cB)); Wire.write(abs (cb));
    Wire.write(button); Wire.write(code);
    Wire.write(klby); Wire.write(klby2);
    Wire.write(klby3); Wire.write(klby4);
    Wire.write(klby); Wire.write(klby2);
    Wire.write(klby3); Wire.write(klby4);
}

```

```
Wire.write(minus2x); Wire.write(minus2y);
Wire.write(abs(sudutk1)); Wire.write(abs(sudutk2));
Wire.write(code2); Wire.write(code3);

}

void data_klbpos() {
  if (ex < 0) minus2x = 0; else minus2x = 1;
  if (ey < 0) minus2y = 0; else minus2y = 1;
  if (sudutk > 180) {
    sudutk1 = 180;
    sudutk2 = sudutk - 180;
  }
  else if (sudutk <= 180) {
    sudutk1 = sudutk;
    sudutk2 = 0;
  }
  }

  klbx = abs(ex);
  if (klbx >= 255) klbx = 255;
  else if (klbx <= 0) klbx = 0;
  else klbx = klbx;
  klbx2 = abs(ex) - 255;
  if (klbx2 >= 255) klbx2 = 255;
  else if (klbx2 <= 0) klbx2 = 0;
  else klbx2 = klbx2;
  klbx3 = abs(ex) - 510;
  if (klbx3 >= 255) klbx3 = 255;
  else if (klbx3 <= 0) klbx3 = 0;
  else klbx3 = klbx3;
  klbx4 = abs(ex) - 765;
  if (klbx4 >= 255) klbx4 = 255;
  else if (klbx4 <= 0) klbx4 = 0;
  else klbx4 = klbx4;

  klby = abs(ey);
  if (klby >= 255) klby = 255;
  else if (klby <= 0) klby = 0;
  else klby = klby;
  klby2 = abs(ey) - 255;
  if (klby2 >= 255) klby2 = 255;
  else if (klby2 <= 0) klby2 = 0;
  else klby2 = klby2;
  klby3 = abs(ey) - 510;
  if (klby3 >= 255) klby3 = 255;
  else if (klby3 <= 0) klby3 = 0;
  else klby3 = klby3;
  klby4 = abs(ey) - 765;
  if (klby4 >= 255) klby4 = 255;
  else if (klby4 <= 0) klby4 = 0;
  else klby4 = klby4;
}
}
```