



**RANCANG BANGUN KURSI RODA OTOMATIS DENGAN HEAD
MOTION CONTROL MENGGUNAKAN SENSOR GYROSCOPE UNTUK
PENYANDANG DISITABILITAS**

SKRIPSI

Oleh

Tigo Haq Nursab

NIM 161910201011

PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS JEMBER

2020



**RANCANG BANGUN KURSI RODA OTOMATIS DENGAN HEAD
MOTION CONTROL MENGGUNAKAN SENSOR GYROSCOPE UNTUK
PENYANDANG DISITABILITAS**

SKRIPSI

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

Oleh

Tigo Haq Nursab

NIM 161910201011

PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS JEMBER

2020

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Allah SWT yang telah memberikan Rahmat dan Hidayah-Nya yang telah memberikan kekuatan, kesehatan dan kesabaran dalam mengerjakan skripsi ini.
2. Junjungan kita Nabi Muhammad SAW sebagai penutan kita dalam segala hal.
3. Kedua Orang Tua saya, Ayahanda Kardosin dan Ibunda Jumiaty tercinta, terimakasih atas semua pengorbanan, usaha, jerih payah, kasih sayang, dorongan, nasehat dan air mata yang menetes dalam setiap untaian doa yang senantiasa mengiringi setiap langkah perjuangan serta keberhasilan penulis.
4. Kedua Kakak saya, Lindy Kimiarsih dan Dovan Hyu Sepfrendy yang mana telah membantu membiayai kuliah dan mensupport penulis dari awal kuliah sampai menyelesaikan skripsi ini.
5. Bapak Dosen Pembimbing Utama Bapak Ir. Khairul Anam, S.T., M.T., Ph.D serta Bapak Dosen Pembimbing Anggota Bapak Arizal Mujibtamala Nanda Imron, S.T., M.T atas kesabaran dan keikhlasan dalam membimbing penulis menyelesaikan skripsi ini.
6. Seluruh dosen Teknik Elektro Universitas Jember yang telah memberikan bekal ilmu pengetahuan dan bimbingan selama mengikuti pendidikan di Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.
7. Almamaterku Universitas Jember yang aku cintai dan banggakan
8. Keluarga Besar UKM Robotika Teknik Universitas Jember sebagai wadah penulis dalam memperoleh ilmu dan wawasan keorganisasian, dan mendapatkan keluarga baru dalam mengemban tugas bersama.
9. Sahabat-sahabat ku tercinta dan dulu – dulur Teknik elektro Angkatan 2016 yang telah memberikan semangat serta perhatian yang luar biasa.
10. Keluarga Besar Peneliti Laboratorium Sistem Cerdas dan Robotika, CDAST serta Asisten Laboratorium Elektronika dan Terapan, Patrang.

MOTTO

“Bersabarlah kamu dan kuatkanlah kesabaranmu dan tetaplah bersiap siaga dan bertaqwalah kepada Allah supaya kamu menang”

(QS. Al Imraan : 200)

“Yakinlah, ada sesuatu yang menantimu setelah banyak kesabaran (yang kau jalani), yang akan membuatmu terpana hingga kau lupa betapa pedihnya rasa sakit”

(Ali Bin Abi Thalib)

“Tuhan tidak menuntut kita untuk sukses. Tuhan hanya menyuruh kita berjuang tanpa henti”

(Emha Ainun Nadjib)

“Orang boleh salah, agar dengan demikian ia berpeluang menemukan kebenaran dengan proses autentiknya sendiri”

(Emha Ainun Nadjib)

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Tigo Haq Nursab

NIM : 161910201011

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul: “**Rancang Bangun Kursi Roda Otomatis Dengan *Head Motion Control* Menggunakan Sensor *Gyroscope* Untuk Penyandang Disabilitas**” adalah benar-benar karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya dan belum pernah diajukan pada institusi mana pun, serta bukan karya jiplakan. Saya bertanggung jawab 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 mana pun serta bersedia mendapat sanksi akademik jika ternyata dikemudian hari pernyataan ini tidak benar.

Jember, 12 Juni 2020

Yang menyatakan,

Tigo Haq Nursab

NIM.161910201011

SKRIPSI

**RANCANG BANGUN KURSI RODA OTOMATIS DENGAN HEAT
MOTION CONTROL MENGGUNAKAN SENSOR GYROSCOPE UNTUK
PENYANDANG DISITABILITAS**

Oleh

Tigo Haq Nursab
NIM 161910201011

Pembimbing

Dosen Pembimbing Utama : Ir. Khairul Anam, S.T., M.T., Ph.D

Dosen Pembimbing Anggota : Arizal Mujibtamala Nanda Imron, S.T., M.T

PENGESAHAN

Skripsi berjudul “Rancang Bangun Kursi Roda Otomatis Dengan *Head Motion Control* Menggunakan Sensor *Gyroscope* Untuk Penyandang Disabilitas” karya Tigo Haq Nursab telah diuji dan disahkan pada:

Hari : Kamis
Tanggal : 02 Juli 2020
Tempat : Fakultas Teknik

Tim Penguji,

Pembimbing Utama

Pembimbing Anggota,

Ir. Khairul Anam, S.T., M.T., Ph.D
NIP 198002072015042001

Arizal Mujibtamala Nanda Imron, S.T., M.T.
NIP 760017099

Penguji I,

Penguji II,

Dr. Ir. Satryo Budi Utomo, ST., MT
NIP 198501262008011002

Ir Ike Fibriani, S.T, M.T
NIP 198002072015042001

Mengesahkan
Dekan,

Dr. Ir. Triwahju Hardianto, S.T., M.T.
NIP 197008261997021001

RINGKASAN

Rancang Bangun Kursi Roda Otomatis Dengan *Head Motion Control* Menggunakan Sensor *Gyroscope* Untuk Penyandang Disabilitas; Tigo Haq Nursab, 161910201011; 2020 ; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Pada umumnya sebagian orang mengetahui bahwa kursi roda komersial bergerak dengan cara di dorong oleh seseorang ataupun bergerak dengan cara di putar dengan bantuan tangan pengguna, selain kursi roda komersial tersebut terdapat juga kursi roda elektrik yang mana bergerak dengan bantuan putaran motor yang di kendalikan pada tuas kendali (*Joystick*) pada bagian tangan kursi roda. Sebagian kursi roda dibutuhkan oleh orang disabilitas yang tidak dapat berjalan menggunakan kakinya namun dapat mengoperasikan kursi roda menggunakan tangan, bagaimana jika orang disabilitas jika tidak dapat mengoperasikan kursi roda menggunakan tangan, pastinya akan membebani orang di sekitar. Dengan demikian penderita disabilitas harus memanfaatkan bagian tubuh yang lain untuk mengoperasikan kursi roda tersebut.

Kerja Kursi Roda otomatis dengan pergerakan kepala menggunakan sensor *Gyroscope* nantinya akan mengendalikan pergerakan kursi roda melalui perintah yang di hasilkan oleh kemiringan pada kepala. Sensor *Gyroscope* yang berada di topi kendali dan Mikrokontroler *Arduino Nano* nantinya akan membaca kemiringan sudut kepala dan mengirimkan hasil baca tersebut ke mikrokontroler STM32F103 yang berada di kursi roda sebagai proses kendali pergerakan motor pada kursi roda menggunakan modul nRF24L01, yang mana jika kepala dimiringkan kebawah (menunduk) akan memerintah kursi roda untuk jalan maju, jika kepala miring kanan kursi roda akan belok ke kanan, jika kepala miring ke kiri kursi roda akan belok kiri dan jika kepala dimiringkan ke atas kursi roda akan berjalan mundur. Dari pergerakan kursi roda tersebut nantinya akan dibaca kecepatan pada roda kursi roda menggunakan sensor *Rotary Encoder*, dan akan menstabilkan kecepatan kursi roda sesuai *set point* yang di tentukan menggunakan metode perhitungan *Proporsional Integral Derivatif* (PID).

Hasil dari penelitian ini pengujian kursi roda dengan kendali pergerakan kepala memiliki prosentase rata - rata keberhasilan pada saat proses pengujian sebesar 96.7%. Dengan nilai tuning PID yang digunakan yaitu tuning menggunakan *Trial and Error* dengan nilai K_p sebesar 0.011, nilai K_i sebesar 0.00011, dan nilai K_d sebesar 7 sehingga menghasilkan nilai *Rise Time* (T_r) sebesar 0.3s, nilai *Setling Time* (T_s) sebesar 1s dan nilai *Error Over Shoot* ($M_p\%$) sebesar 7.1%.



SUMMARY

Design an Automatic Wheelchairs with Movement Control Heads Using Gyroscope Sensors for People with Disabilities; Tigo Haq Nursab, 161910201011; 2020; Department of Electrical Engineering Faculty of Engineering University of Jember

In general, some people know that commercial wheelchairs move by being pushed by someone or move by being swiveled with the help of the user's hand, in addition to these commercial wheelchairs there are also electric wheelchairs which move with the aid of motor rotation controlled on the control lever (Joystick) on the wheelchair hand. Some wheelchairs are needed by people with disabilities who cannot walk using their feet but can operate a wheelchair using their hands, what if a person with a disability cannot operate a wheelchair using their hands, it will certainly burden people around. Thus, people with disabilities must utilize other body parts to operate the wheelchair.

Automatic Wheelchairs Work with head movement using a Gyroscope sensor will later control the movement of the wheelchair through commands generated by the tilt of the head. Gyroscope sensor which is in the control cap and Arduino Nano Microcontroller will later read the tilt angle of the head and send the readings to the STM32F103 microcontroller in the wheelchair as the process of controlling motor movement in the wheelchair using the nRF24L01 module, which if the head is tilted down (downward)) will command the wheelchair to go forward, if the head is tilted right the wheelchair will turn right, if the head is tilted to the left the wheelchair will turn left and if the head is tilted upward the wheelchair will walk backwards. From the movement of the wheelchair the speed of the wheelchair will be read using a Rotary Encoder sensor, and it will stabilize the wheelchair speed according to the set point determined using the Proportional Integral Derivative (PID) calculation method.

The results of this study wheelchair testing with head movement control has an average percentage of success during the testing process of 96.7%. With the PID tuning value used is tuning using Trial and Error with a K_p value of 0.011, a K_i value of 0.00011, and a K_d value of 7 resulting in a Rise Time (T_r) value of 0.3s, a Settling Time (T_s) value of 1s and a value of Error Over Shoot ($M_p\%$) by 7.1%.



PRAKATA

Puji syukur ke hadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “**Rancang Bangun Kursi Roda Otomatis Dengan *Head Motion Control* Menggunakan Sensor Gyroscope Untuk Penyandang Disabilitas**”. 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 saya dapat menuntaskan skripsi ini.
2. Nabi Muhammad SAW, Nabi junjungan umat Islam yang telah menarik kami dari zaman jahiliyah ke zaman yang lebih baik.
3. Kepada Pahlawan dalam Kehidupan saya selama ini, bapak Kardosin, ibu Jumiati, Kedua kakak saya Lindy Kimiarsih dan Dovan Hyu Sepfrendy yang telah membantu baik moral atau materil serta doa yang tidak pernah putus.
4. Bapak Ir. Khairul Anam, S.T., M.T., Ph.D. selaku Dosen Pembimbing Utama dan Bapak Arizal Mujibtamala Nanda Imron, S.T., M.T. selaku Dosen Pembimbing Anggota yang telah meluangkan waktu dan fikiran serta perhatiannya guna memberikan bimbingan dan arahan demi terselesainya skripsi ini.
5. Bapak Dr. Ir. Satryo Budi Utomo S.T., M.T. selaku dosen penguji utama dan Ibu Ir. Ike Fibriani, S.T., M.T. selaku dosen penguji anggota yang telah memberikan kritik dan saran yang membangun sehingga sangat membantu terhadap penyempurnaan skripsi ini.
6. Bapak Alfredo Bayu Satrya S.T., M.T. selaku dosen pembimbing akademik yang telah membimbing dan menanamkan rasa disiplin dan tanggung jawab dengan apa yang dilakukan selama penulis menjadi mahasiswa.
7. Seluruh Dosen Teknik Elektro Universitas Jember yang telah memberikan ilmu pengetahuan dan bimbingan selama mengikuti pendidikan di Universitas Jember

8. Guru-guru yang telah membimbing saya mulai dari taman kanak-kanak sampai sekolah menengah.
9. Keluarga Teknik Elektro 2016 “INDUKTRO” dan teman teman konsentrasi “Elektronika’16” yang sudah menemani saya berproses selama menjadi mahasiswa Teknik Elektro Universitas Jember.
10. Teman seperjuangan se-DPU, se-DPA dan sepenelitian di Laboratorium Sistem Cerdas dan Robotika, CDAST, serta Laboratorium Elektronika dan Terapan, Patrang yang saling mendukung, saling membantu, saling menyemangati dan memotivasi dalam penyusunan skripsi.
11. Teman – teman penghuni LPM Universitas Jember terimakasih telah banyak membantu setiap masalah yang saya hadapi dalam menyelesaikan skripsi saya.
12. Teman-teman Asisten 2016 Laboratorium Teknik “Patrang” Universitas Jember sudah menemani selama saya di Laboratorium.
13. Christine Fatmasari yang selalu menyemangati dikala lagi malas dan selalu mendengarkan keluh kesah saya.
14. 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 supaya informasi dari skripsi ini dapat memberikan manfaat bagi penulis dan pembaca

Jember, 06 juni 2020

Penulis

DAFTAR ISI

| | |
|---|----------|
| HALAMAN SAMPUL..... | i |
| HALAMAN JUDUL | ii |
| HALAMAN PERSEMBAHAN | iii |
| HALAMAN MOTTO | iv |
| HALAMAN PERNYATAAN..... | v |
| HALAMAN PEMBIMBING | vi |
| HALAMAN PENGESAHAN..... | vii |
| RINGKASAN | viii |
| SUMMARY | x |
| PRAKATA | xii |
| DAFTAR ISI..... | xiv |
| DAFTAR GAMBAR..... | xvii |
| DAFTAR TABEL | xx |
| BAB 1 PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah..... | 3 |
| 1.3 Tujuan Penelitian | 3 |
| 1.4 Batasan Masalah | 3 |
| 1.5 Manfaat Penelitian | 3 |
| BAB 2 TINJAUAN PUSTAKA..... | 4 |
| 2.1 Kursi Roda..... | 4 |
| 2.2 Arduino Nano | 5 |
| 2.3 STM 32F103 (Bluepill)..... | 6 |

| | | |
|--|--|-----------|
| 2.4 | Driver BTS 7960 | 7 |
| 2.5 | Motor DC | 8 |
| 2.6 | Sensor Gyroscope | 8 |
| 2.7 | Modul nRF 24L01 | 9 |
| 2.8 | Rotary Encoder | 10 |
| 2.9 | Kontrol PID | 11 |
| BAB 3 METODE PELAKSANAAN DATA | | 15 |
| 3.1 | Tempat dan Waktu Penelitian | 15 |
| 3.2 | Tahapan Penelitian | 15 |
| 3.3 | Alat dan Bahan | 16 |
| 3.4 | Perancangan mekanik Kursi Roda dan Topi Kendali | 17 |
| 3.4.1 | Perancangan Mekanik Kursi Roda..... | 17 |
| 3.4.2 | Perancangan Mekanik Topi Kendali..... | 18 |
| 3.5 | Perancangan Sistem Elektronika Kursi Roda dan Topi Kendali... | 18 |
| 3.5.1 | Perancangan Sistem Elektronika Topi Kendali..... | 18 |
| 3.5.2 | Perancangan Sistem Elektronika Kursi Roda..... | 20 |
| 3.6 | Perancangan Perangkat Lunak | 22 |
| 3.6.1 | Perancangan Perangkat Lunak Secara Keseluruhan..... | 22 |
| 3.6.2 | Perancangan Perangkat Lunak Pada Topi Kendali..... | 23 |
| 3.6.3 | Perancangan Kontrol dan Tuning PID..... | 24 |
| 3.7 | Pola Gerak Kendali Pada Kepala | 24 |
| 3.8 | Design Pengambilan dan pengujian Data | 25 |
| 3.8.1 | Data Transfer Data nRF24L01..... | 25 |
| 3.8.2 | Data Gyroscope..... | 25 |
| 3.8.3 | Pengujian Motor..... | 26 |
| 3.8.4 | Pengujian kontrol PID..... | 26 |
| 3.8.5 | Pengujian Keseluruhan..... | 26 |

| | |
|---|-----------|
| 3.8.6 Hasil Perancangan Alat | 26 |
| BAB 4. HASIL DAN PEMBAHASAN | 27 |
| 4.1 Kalibrasi dan Pengujian Sensor Gyroscope MPU6050 | 29 |
| 4.1.1 Kalibrasi Sensor Gyroscope MPU6050 | 29 |
| 4.1.2 Pengujian Sensor Gyroscope MPU6050 | 35 |
| 4.2 Modul nRF24L01 | 36 |
| 4.2.1 Pengujian Konektivitas Modul nRF24L01 | 36 |
| 4.2.2 Pengujian Pengiriman Data Modul nRF24L01 | 37 |
| 4.2.3 Pengujian Waktu Respon Pengiriman dan Penerimaan Instruksi | 40 |
| 4.3 Pengujian Rotary Encoder Motor | 41 |
| 4.3.1 Pengujian Rotary Encoder Motor Kiri | 41 |
| 4.3.2 Pengujian Rotary Encoder Motor Kanan | 42 |
| 4.4 Kontrol PID | 44 |
| 4.4.1 Tuning Parameter PID secara Manual | 44 |
| 4.4.2 Tuning Parameter PID dengan Ziegler-Nichols II (Close Loop) | 47 |
| 4.4.3 Pengujian Kontrol PID | 50 |
| 4.5 Pengujian Keseluruhan | 54 |
| BAB 5. KESIMPULAN DAN SARAN | 65 |
| 5.1 Kesimpulan | 65 |
| 5.2 Saran | 65 |
| DAFTAR PUSTAKA | 66 |
| LAMPIRAN | 67 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2. 2 a) Konvensional wheelchair b) Platform model | 4 |
| Gambar 2. 3 <i>Arduino Nano</i> | 5 |
| Gambar 2. 4 Pin <i>Arduino Nano</i> | 5 |
| Gambar 2. 5 Pin Out <i>STM 32F103 (BluePill)</i> | 7 |
| Gambar 2. 6 Modul Driver <i>BTS 7960</i> | 7 |
| Gambar 2. 7 Sistem pada Motor DC | 8 |
| Gambar 2. 8 bentuk fisik sensor <i>gyroscope</i> | 9 |
| Gambar 2. 9 Modul <i>nRF24L01</i> | 10 |
| Gambar 2. 10 Desain <i>Rotary Encoder</i> | 10 |
| Gambar 2. 11 Diagram sistem kontrol <i>PID</i> | 11 |
| Gambar 2. 12 Pengaruh Konstanta K_p Terhadap Respon Kontrol <i>PID</i> | 12 |
| Gambar 2. 13 Pengaruh konstanta K_i terhadap respon kontrol <i>PID</i> | 13 |
| Gambar 2. 14 pengaruh K_d terhadap response <i>PID</i> | 13 |
| Gambar 2. 15 Osilasi berkesinambungan dengan periode P_{cr} | 14 |
| Gambar 3. 1 <i>Design</i> Kursi Roda | 17 |
| Gambar 3. 2 <i>Design</i> Topi Kendali | 18 |
| Gambar 3. 3 Diagram Blok sistem elektronika Kendali | 19 |
| Gambar 3. 4 Sistem Elektronika Topi Kendali | 19 |
| Gambar 3. 5 (a) Layout Elektronika Topi Kendali (b) Schematic Elektronika Topi Kendali. | 19 |
| Gambar 3. 6 Diagram Blok Sistem Elektronika Kursi Roda | 20 |
| Gambar 3. 7 Sistem Elektronika Kursi Roda | 20 |
| Gambar 3. 8 (a) <i>Layout</i> elektronika Kursi Roda (b) <i>Schematic</i> Kursi Roda..... | 21 |
| Gambar 3. 9 Kinerja rangkaian topi kendali dan rangkaian kursi roda | 21 |
| Gambar 3. 10 Diagram Alir Alat Keseluruhan | 22 |
| Gambar 3. 11 Diagram Alir Alat Bagian Topi Kendali | 23 |
| Gambar 3. 12 Diagram blok kontrol <i>PID</i> Motor Kursi Roda | 24 |
| Gambar 3. 13 Posisi Awal Kepala (a) Tampak Depan (b) Tampak Samping | 24 |
| Gambar 3. 14 Pola Pergerakan (a) Belok Kanan-Kiri, dan (b) Maju-Mundur | 25 |
| Gambar 3. 15 (a) Kursi Roda (b) kendali kursi roda..... | 26 |

| | |
|---|----|
| Gambar 4. 1 Mekanik kursi Roda | 27 |
| Gambar 4. 2 Rangkain pada Topi Kendali | 28 |
| Gambar 4. 3 Elektrikal Kontrol Motor pada Kursi Roda..... | 28 |
| Gambar 4. 4 Ilustrasi Kinerja Alat. | 29 |
| Gambar 4. 5 Grafik nilai regresi sudut sumbu x | 32 |
| Gambar 4. 6 Grafik nilai regresi sudut sumbu y | 34 |
| Gambar 4. 7 Pengujian sensor <i>Gyroscope</i> MPU6050..... | 36 |
| Gambar 4. 8 Pengiriman Data Instruksi Berhenti | 38 |
| Gambar 4. 9 Pengiriman Data Instruksi Kanan | 38 |
| Gambar 4. 10 Pengiriman Data Instruksi Kiri | 39 |
| Gambar 4. 11 Pengiriman Data Instruksi Maju | 39 |
| Gambar 4. 12 Pengiriman Data Instruksi Mundur | 39 |
| Gambar 4. 13 <i>Rotary Encoder</i> pada Kursi Roda..... | 41 |
| Gambar 4. 14 Grafik perbandingan Kecepatan Motor Kiri | 42 |
| Gambar 4. 15 Grafik perbandingan Kecepatan Motor Kanan | 43 |
| Gambar 4. 16 Pengambilan data <i>Rotary Encoder</i> | 43 |
| Gambar 4. 17 Grafik Percobaan Pertama <i>Tunning</i> PID..... | 44 |
| Gambar 4. 18 Grafik Percobaan Kedua <i>Tunning</i> PID..... | 45 |
| Gambar 4. 19 Grafik Percobaan Ketiga <i>Tunning</i> PID | 45 |
| Gambar 4. 20 Grafik Percobaan Keempat <i>Tunning</i> PID..... | 46 |
| Gambar 4. 21 Grafik perbandingan hasil <i>Tunning</i> PID | 46 |
| Gambar 4. 22 Grafik osilasi <i>tunning</i> Kp <i>Ziegler-Nichols 2</i> | 47 |
| Gambar 4. 23 Grafik hasil <i>Ziegler-Nichols 2</i> | 48 |
| Gambar 4. 24 Grafik hasil <i>tunning</i> <i>Ziegler-Nichols 2</i> setelah diubah..... | 49 |
| Gambar 4. 25 Grafik pengujian 40 RPM didalam ruangan | 50 |
| Gambar 4. 26 Grafik pengujian 60 RPM didalam ruangan | 51 |
| Gambar 4. 27 Pengujian didalam ruangan | 51 |
| Gambar 4. 28 Grafik pengujian 40 RPM diluar ruangan | 52 |
| Gambar 4. 29 Grafik pengujian 60 RPM diluar ruangan..... | 52 |
| Gambar 4. 30 Pengujian diluar ruangan..... | 53 |

| | |
|--|----|
| Gambar 4. 31 Grafik pengujian 40 RPM di tanjakan beraspal | 53 |
| Gambar 4. 32 Grafik pengujian 60 RPM di tanjakan beraspal | 54 |
| Gambar 4. 33 Pengujian di tanjakan | 54 |
| Gambar 4. 34 Pengujian Keseluruhan Rintangan Pertama | 55 |
| Gambar 4. 35 Pengujian Keseluruhan Rintangan Kedua..... | 58 |
| Gambar 4. 36 Uji Coba Responden Pertama | 62 |
| Gambar 4. 37 Uji Coba Responden Kedua | 62 |
| Gambar 4. 38 Uji coba Responden Ketiga..... | 62 |
| Gambar 4. 39 Uji kenyamanan kursi roda oleh responden hari pertama. | 63 |
| Gambar 4. 40 Uji kenyamanan Kursi Roda oleh responden hari kedua | 64 |

DAFTAR TABEL

| | |
|---|----|
| Tabel 2. 1 Spesifikasi Arduino Nano | 6 |
| Tabel 2. 2 Aturan Dasar <i>Ziegler-Nichols</i> Berdasarkan <i>Kcr</i> dan <i>Pcr</i> | 14 |
| Tabel 3. 1 Rencana dan Jadwal Pelaksanaan Penelitian | 15 |
| Tabel 4. 1 Pengambilan Sampel Sudut Sumbu x | 31 |
| Tabel 4. 2 Nilai kalibrasi sudut pada sumbu x | 32 |
| Tabel 4. 3 Pengambilan Sampel Sudut pada Sumbu y | 33 |
| Tabel 4. 4 Nilai kalibrasi sudut pada sumbu y | 34 |
| Tabel 4. 5 Pengujian sensor <i>Gyroscope</i> MPU6050 | 35 |
| Tabel 4. 6 Pengujian modul nRF24L01 dalam ruangan | 37 |
| Tabel 4. 7 Pengujian modul nRF24L01 luar ruangan | 37 |
| Tabel 4. 8 Pengujian waktu respon pengiriman dan penerimaan instruksi..... | 40 |
| Tabel 4. 9 Data Pengujian <i>Rotari Encoder</i> Motor Kiri..... | 42 |
| Tabel 4. 10 Data Pengujian <i>Rotari Encoder</i> Motor Kanan..... | 43 |
| Tabel 4. 11 <i>Tunning</i> Parameter PID..... | 44 |
| Tabel 4. 12 Perbandingan kontrol PID <i>tunning</i> manual dan <i>Ziegler-Nichols</i> | 49 |
| Tabel 4. 13 Percobaan pertama responden pertama memutari halangan | 56 |
| Tabel 4. 14 Percobaan pertama responden kedua memutari halangan | 56 |
| Tabel 4. 15 Percobaan pertama responden ketiga memutari halangan | 57 |
| Tabel 4. 16 Percobaan kedua responden pertama melewati halangan | 59 |
| Tabel 4. 17 Percobaan responden kedua kedua melewati halangan | 60 |
| Tabel 4. 18 Percobaan kedua responden ketiga melewati halangan | 61 |
| Tabel 4. 19 Pengujian hari pertama..... | 63 |
| Tabel 4. 20 Pengujian hari kedua | 64 |

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Penyakit kelumpuhan atau paralisis menurut bahasa medis, merupakan gangguan dari beberapa fungsi pergerakan tubuh yang tidak dapat di gerakan. Kelumpuhan pada umumnya bersifat sementara maupun permanen, tergantung dengan penyebab terjadi-nya kelumpuhan. Adapun beberapa penyebab kelumpuhan yang terjadi antara lain dikarenakan faktor kelainan sejak lahir, faktor usia, pasca kecelakaan, ataupun sakit karena komplikasi, misalnya *stroke*, ataupun darah tinggi. Para penderita kelumpuhan dengan disabilitas yang parah biasanya tidak dapat melakukan kegiatan sehari – hari, seperti berjalan dan bergerak menyusuri ruang dengan baik tanpa adanya alat bantu perangkat medis seperti tongkat maupun kursi roda.

Secara umum kursi roda bergerak menggunakan bantuan tangan atau menggunakan bantuan perangkat elektrik berupa motor, tergantung pengguna kursi roda. Jika keadaan pengguna kursi roda mengalami kelumpuhan pada bagian tangan maupun kaki, bergerak menggunakan bantuan tangan sendiri sulit jika dilakukan, dan harus meminta bantuan saudara ataupun orang terdekat untuk mendorong kursi roda. Hal tersebut tentunya sangat membebani orang di sekitar, berbeda dengan kursi roda elektrik yang bisa bergerak tanpa bantuan orang lain melainkan dengan bantuan motor untuk memutar roda agar kursi roda bergerak. Meskipun pada kursi roda elektrik masih terdapat kekurangan yang sama yaitu ketika tangan seorang yang lumpuh tidak dapat di gerakan untuk menggerakkan tuas pengendali kursi roda.

Dengan adanya perkembangan teknologi yang sangat pesat, khususnya di bidang Robotika maupun di bidang Medis, sulitnya penggunaan kursi roda bagi penderita lumpuh atau disabilitas pada bagian kaki maupun pada bagian tangan telah banyak dilakukan penelitian. Terdapat beberapa solusi yang telah dilakukan dan dikembangkan, salah satunya yaitu penelitian yang berjudul “*Wheel Chair Control By Head Motion*” (Pajkanovic dan Dokic, 2013), yang mana membuat sebuah penelitian kursi roda otomatis menggunakan sensor *accelerometer*, untuk

mendeteksi pergerakan kepala sebagai pengendali jalannya kursi roda. Dalam penelitian mereka berdua masih menggunakan mikrokontroler ATMEGA 16, dan sensor *accelerometer* ADXL330, dengan teorema *Low Pass Filter* sebagai penyetabil sinyal yang dihasilkan oleh pembacaan sensor *accelerometer*. Selain itu juga terdapat tambahan perantara penggerak berupa mekanik penggerak yang terbuat dari aluminium dan *servo* untuk mengendalikan *joystick* pada kursi roda.

Selain penelitian yang dilakukan oleh Aleksandar Pajkanovic dan rekannya Branco Dokic, terdapat juga penelitian kursi roda lain yang berjudul “*The Smart Wheelchair Component System*”(Simpson dkk., 2004). Dimana pada penelitian mereka di jelaskan bahwa kursi roda menggunakan beberapa sensor untuk navigasi berupa sensor ultrasonik, sensor *infrared* serta beberapa sensor lain untuk jalan-nya robot, serta menggunakan bahasa C++ sebagai bahasa pemrogramannya. Dari penelitian di atas masih terdapat beberapa kekurangan antara lain, kurang efisiennya sistem navigasi pada pergerakan kursi roda yang menggunakan banyak sensor.

Sehingga pada penelitian ini penulis akan meriset Kursi Roda Otomatis dengan pembacaan pergerakan kepala menggunakan sensor *gyroscope* serta *stm32f103* tanpa adanya mekanik tambahan untuk mengendalikan *joystick* pada kursi roda. Di mana pada penelitian ini ada dua bagian alat yang akan di buat yaitu bagian topi yang digunakan untuk tempat sensor *gyroscope* sebagai pembacaan kemiringan kepala atau kendali pada kursi roda. Pada topi ini terdapat *Arduino Nano*, Sensor *gyroscope*, Modul *transmitter*, dan *battery* sebagai catu daya. Selain perancangan bagian Topi sebagai kendali, pada penelitian ini juga merancang bagian kursi roda itu sendiri, yang tersusun atas beberapa komponen antara lain badan struktur kursi roda, mikrokontroler, motor DC *driver* motor, Modul *reiciver*, dan tentunya Catu Daya. Kedua bagian ini antara topi kendali dan kursi roda nantinya akan saling berkomunikasi secara nirkabel agar kursi roda bergerak sesuai instruksi pergerakan kepala yang telah di terjemahkan oleh sensor dan mikrokontroler. Dengan diterapkannya penelitian ini diharapkan para penderita disabilitas lebih ter-bantu dalam beraktifitas sehari hari tanpa membutuhkan seseorang untuk menggerakkan kursi roda.

1.2 Rumusan Masalah

Dari latar belakang di atas terdapat beberapa masalah antara lain:

- a. Bagaimana cara membuat kendali kursi roda berbasis pergerakan kepala untuk kursi roda elektrik?
- b. Bagaimana cara merancang kontrol PID pada motor kursi roda agar pergerakan kursi roda stabil?
- c. Bagaimana cara menguji kursi roda pada pengguna kursi roda.

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka penelitian ini dilakukan dengan tujuan sebagai berikut:

- a. Dapat membuat sebuah kursi roda otomatis dengan instruksi pergerakan kepala.
- b. Dapat membuat kontrol kestabilan pada respon motor kursi roda dengan menggunakan kontrol PID.
- c. Dapat menguji kegunaan kursi robot dengan pengguna kursi roda.

1.4 Batasan Masalah

Untuk mempermudah dan terfokus pada dalam penelitian maka ada beberapa batasan masalah yang di antaranya yaitu sebagai berikut:

- a. Pada penelitian ini hanya menggunakan dua mikrokontroler yang berbasis *Arduino*.
- b. Target dari penelitian kursi roda dapat berjalan menggunakan instruksi pembacaan sensor *gyroscope*.
- c. Komunikasi antar mikrokontroler menggunakan *nRF 24L01*
- d. Pada motor menggunakan kontrol PID.
- e. Motor pada kursi roda menggunakan motor bawaan dari kursi roda.
- f. Instruksi pada jalannya kursi roda hanya jalan maju, jalan mundur, belok kanan dan belok kiri.

1.5 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah:

- a. Diharapkan alat ini dapat membantu para penderita disabilitas.
- b. Menjadi inovasi di dunia Robotika khususnya di bidang Medis.

BAB 2 TINJAUAN PUSTAKA

Pada bab ini menjelaskan mengenai beberapa bagian alat baik komponen maupun metode yang menjadi dasar dalam proses mendesign maupun pembuatan kursi roda otomatis untuk meningkatkan kinerja alat guna penelitian di tahun berikutnya. Hal yang pertama yang di bahas pada bab ini yaitu mengenai kursi roda, dan beberapa metode yang akan digunakan , kemudian juga akan membahas mengenai komponen - komponen yang digunakan seperti *STM32F103*, Arduino dan beberapa sensor maupun aktuator seperti Sensor *gyroscope* dan motor sebagai penggerak.

2.1 Kursi Roda

Kursi roda (*wheelchair*) merupakan alat bantu bagi penyandang disabilitas untuk dapat berpindah dari satu tempat ke tempat lain, baik di dalam ruangan ataupun ruang terbuka seperti jalan raya maupun taman. Pada umumnya kursi roda digunakan untuk meningkatkan mobilitas bagi penyandang disabilitas seperti, orang yang memiliki kekurangan pada fisik (khususnya kaki), pasien rumah sakit yang tidak diperbolehkan untuk melakukan banyak aktivitas fisik, manula, dan orang yang memiliki resiko tinggi untuk terluka, bila berjalan sendiri.

Secara umum kursi roda dibagi menjadi dua jenis, yaitu kursi roda manual (*conventional wheelchair*) dan kursi roda dengan putaran motor (*motor powered wheelchair*). Jenis manual terbagi menjadi kursi roda *standard* dan *sport wheelchair*. Sedangkan kursi roda dengan putaran motor dibagi menjadi beberapa model antara lain, *traditional, platform, dan round based model*. Gambar 2.1 menunjukkan 2 buah kursi roda, yang konvensional dan bantuan putaran motor.



Gambar 2. 1 a) *Konvensional wheelchair* b) Platform model

(Batan, 2006)

Berikut spesifikasi dari *Arduino Nano* :

Tabel 2. 1 Spesifikasi Arduino Nano

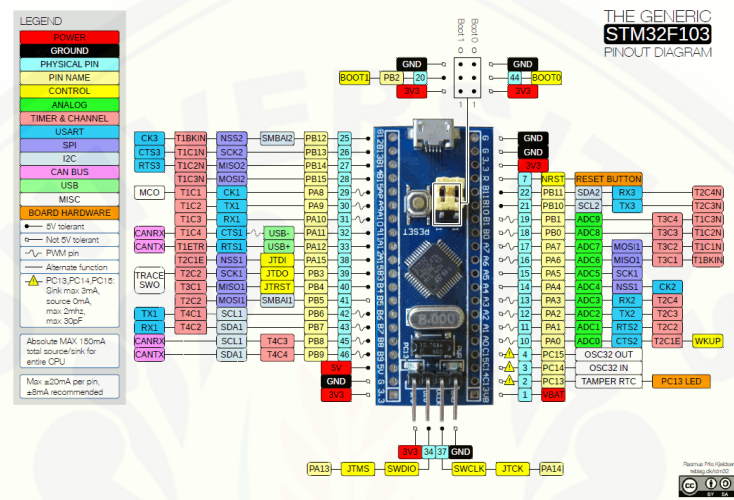
| | |
|--------------------------------------|--|
| Mikrokontroler | Atmel ATmega168 atau ATmega328 |
| Tegangan Operasi | 5 V |
| Input Voltage (disarankan) | 7-12V |
| Input Voltage (limit) | 6-20V |
| Pin Digital I/O | 14 (6 pin digunakan sebagai output PWM) |
| Pins Input Analog | 8 |
| Arus DC per pin I/O | 40mA |
| Flash Memory | 16KB (ATmega168) atau 32KB (ATmega328) |
| SRAM | 1 KB (ATmega168) atau 2 KB (ATmega328) |
| EEPROM | 512 byte (ATmega168) atau 1KB (ATmega328) |
| Clock Speed | 16 MHz |
| Ukuran | 1.85cm x 4.3cm |

2.3 STM 32F103 (Bluepill)

STM32F103C8T6 merupakan seri dari mikrokontroler *STM32 ARM Cortex- M3* yang masuk dalam tipe mainstream. *STM32F103C8T6* memiliki processor 32-bit *ARM Cortex-M3* serta clock frekuensi 72 MHz. Mikrokontroler *STM32* berbeda dengan *ATmega* yang tersedia dalam bentuk DIP (*Dual In Line Package*) yang mudah dalam pemasangannya, melainkan hanya tersedia dalam bentuk LQFP (*Low Profile Quad Flat Package*) atau dalam bentuk SMD yang pemasangan biasanya harus menggunakan solder uap. Terdapat beberapa *board minimum system* mikrokontroler *STM32*, salah satu dari *board* tersebut merupakan hasil produksi diluar produk *STMicroelectronics*. Didalam *board* tersebut terdapat seri *STM32F103C8T6* yang biasa disebut dengan *Bluepill*.

Dengan adanya *board* ini dapat memudahkan pengguna mikrokontroler *STM32* karena tidak perlu lagi membuat *board minimum system* sendiri.

Perlu di ingat bahwa mikrokontroler *STM32* hanya membutuhkan sumber tegangan 2-3,6 v jadi perhatikan pin yang akan digunakan dengan baik. Berikut ini gambar 2.4 merupakan Pinout diagram board Bluepill atau STM32F103C8T6.



Gambar 2. 4 Pin Out STM 32F103 (*BluePill*)

(Hoch dkk., 2018)

2.4 Driver BTS 7960

Pada *driver* motor DC ini dapat mengeluarkan arus hingga 43A, dengan memiliki fungsi PWM. Tegangan sumber DC yang dapat diberikan antara 5.5V-27VDC, sedangkan tegangan *input* antara 3.3V-5VDC, *driver* motor ini menggunakan rangkaian *full H-bridge* dengan IC BTS7960 dengan perlindungan saat terjadi panas dan arus berlebihan. Berikut gambar 2.5 merupakan gambar fisik dari modul Driver BTS 7960.



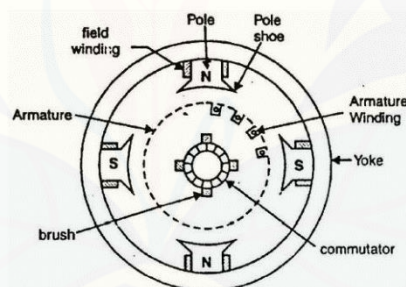
Gambar 2. 5 Modul Driver BTS 7960

(Pradito, 2010)

2.5 Motor DC

Pada prinsip kerja dasar dari sebuah motor *Direct Current* (DC), sebuah batang tembaga yang dapat berotasi bebas dalam medan sebuah magnet permanen. Ketika sebuah arus melalui kumparan, maka menghasilkan medan magnet yang kemudian menimbulkan gaya gerak sehingga menyebabkan rotasi, hal ini terus berlanjut, kumparan berada pada posisi tegak lurus dengan arah arus yang melalui kumparan yang telah di *reverse*.

Pada motor DC konvensional, kumparan tembaga terpasang pada slot sebuah bahan magnetis silinder yang disebut dengan *armature*. *Armature* terpasang pada *bearing*, dan hal ini menyebabkan *armature* dapat berotasi secara bebas. *Armature* ini berada dalam medan magnet yang dihasilkan oleh kutub magnet. Untuk motor yang kecil, magnet permanen atau elektromagnet dengan medan magnet yang dimilikinya dihasilkan oleh sebuah arus yang melalui kumparan. Berikut gambar 2.6 merupakan gambar dari sistem pada Motor DC.



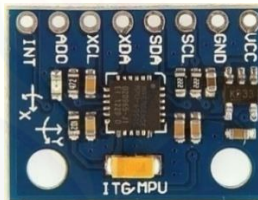
Gambar 2. 6 Sistem pada Motor DC
(Pradito, 2010)

2.6 Sensor Gyroscope

Modul MPU-6050 merupakan modul yang tersusun dengan *chip* berupa MPU-6050 yang memiliki 6 sumbu MPU (*Motion Processing Unit*) serta terdapat pengatur tegangan dan beberapa komponen penyusun lainnya yang membuat modul ini siap pakai dengan tegangan *supply* sebesar 3 -5v. modul MPU-6050 berisi sebuah *accelerometer* dan sebuah *Gyroscope* yang saling terintegrasi. Modul ini dilengkapi dengan fasilitas 16 bit *Analog Digital Converter* setiap bagiannya. Sensor ini akan membaca nilai bagiannya seperti sumbu X, Y, dan Z bersamaan dalam satu waktu.

Berikut merupakan spesifikasi dari modul MPU-6050, dan gambar 2.7 merupakan gambar fisik dari modul MPU6050 :

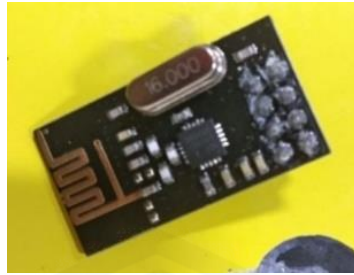
1. Menggunakan *Chip* MPU-6050.
2. *Input* tegangan 2.3 – 3.46 V, akan tetapi pada modul ini terdapat pengatur tegangan sehingga dapat langsung digunakan pada 5v
3. Toleransi guncangan hingga 10000 *g-force*
4. Terdapat *buffer memory* sebesar 1kb.
5. Menggunakan protokol I2C sebagai kendali dan pengumpulan data dengan kecepatan 400 kHz. Pada modul ini juga dilengkapi rangkaian *pull-up* sehingga dapat di sambungkan ke pin SDA SCL secara langsung ke Arduino
6. Rentang skala pada *gyroscope* sebesar 250 derajat, 500 derajat, 1000 derajat, 2000 derajat per detik.
7. Rentang skala accelerometer sebesar 2 *g-force*, 4 *g-force*, 8 *g-force*, dan 16 *g-force*.
8. Konsumsi arus *gyroscope* sebesar 3.4 mA, dan *accelerometer* sebesar 3.8 mA.



Gambar 2. 7 bentuk fisik sensor *gyroscope*
(Fitrianda, 2013a)

2.7 Modul nRF 24L01

nRF24L01 merupakan modul yang dapat berkomunikasi jarak jauh tanpa adanya kabel penghubung (nirkable) dengan frekuensi gelombang radio sebesar 2.4-2.5 GHZ. Kecepatan yang dimiliki modul ini sebesar 250 Kbps, 1 Mbps sampai dengan 2 Mbps. NRF berkomunikasi SPI sebagai antarmuka. Modul ini membutuhkan arus supply sebesar 12.3 mA sebagai mode RX atau penerima. Berikut gambar 2.8 bentuk fisik dari modul nRF24L01.

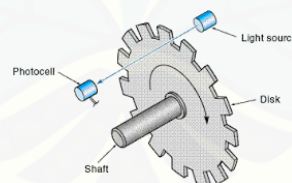


Gambar 2. 8 Modul nRF24L01

(Fitrianda, 2013b)

2.8 Rotary Encoder

Rotary encoder merupakan modul yang dapat difungsikan untuk mengukur dan menghitung suatu Gerakan dan posisi dari putaran. Modul sensor ini memiliki tiga komponen penyusun diantaranya piringan berongga, modul *transmitter* dan *receiver*. Piringan berongga tersebut memiliki banyak rongga atau dapat juga bergerigi sehingga modul *transmitter* berupa sinar dapat melalui rongga pada piringan tersebut untuk dikirim ke *receiver*. Modul *transmitter* tersusun dari beberapa komponen seperti LED sedangkan *Receiver* tersusun dari *photodiode* ataupun LDR sebagai penerima sinar kiriman dari *transmitter*. Berikut gambar 2.9 susunan *transmitter*, *receiver* dan piringan berongga

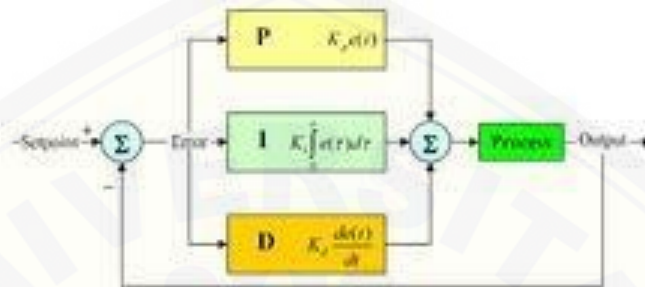
Gambar 2. 9 Desain *Rotary Encoder*

(Fitrianda, 2013a)

Prinsip kerja dari *rotary encoder* ini yaitu menghitung jumlah tangkapan cahaya yang di terima oleh *receiver*. Fungsi dari piringan tersebut yaitu guna untuk menghambat pengiriman cahaya yang dikirim oleh *transmitter* ke *receiver*. Ketika *transmitter* mengirim sinar melewati rongga dan dapat di tangkap oleh *receiver* maka akan menghasilkan pulsa dan jika pengiriman terhalang oleh gerigi maka tidak menghasilkan pulsa pada modul tersebut. Semakin banyak gerigi atau rongga dalam piringan perhitungan data yang dihasilkan semakin akurat.

2.9 Kontrol PID

Kontrol PID merupakan kontrol yang sering digunakan dalam dunia industri. Kontrol PID biasanya digunakan untuk pengontrolan motor listrik yang digunakan untuk mengontrol *konveyor*, pompa untuk pengeboran minyak, dan lain-lain. Berikut gambar 2.10 merupakan diagram sistem dari kontrol PID.



Gambar 2. 10 Diagram sistem kontrol PID
(Fitrianda, 2013a)

PID merupakan singkatan dari *Proportional – Integral – dan Derivative*. Dari tiga pengendali *Proportional – Integral – dan Derivative* yang selanjutnya di jumlahkan menjadi variable yang lain, untuk menghitung keluaran dari nilai kontroler PID. Dengan mendefinisikan $u(t)$ sebagai keluaran PID, maka berikut ini persamaan dari algoritma PID :

$$u(t) = P(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \dots\dots\dots (2.1)$$

Dengan Keterangan sebagai berikut :

$$e = \text{error} = Y_{sp} - Y_m$$

$$Y_{sp} = \text{set point}$$

$$Y_m = \text{Variabel Proses}$$

$$t = \text{waktu}$$

τ = variable integral, nilainya diambil dari waktu nol sampai t.

K_p = gain proporsional, parameter tuning

K_i = gain integral, parameter tuning

K_d = gain derivatif, parameter tuning

Berikut merupakan *Transfer Function* kontrol PID dalam bentuk *laplace* yang mana s merupakan frekuensi bilangan kompleks.

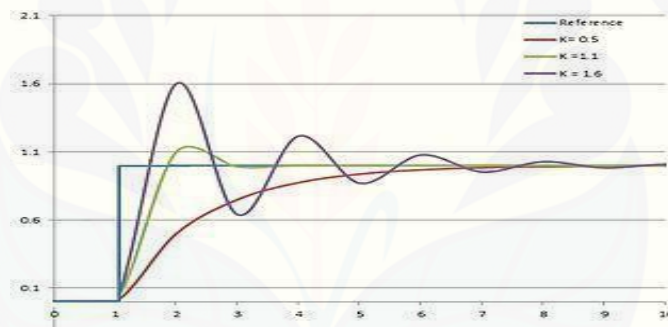
$$L(s) = K_p + K_i/s + K_d s \dots\dots\dots(2.2)$$

Proporsional menghasilkan nilai output dengan nilai berbanding lurus dengan nilai *error*. Respon dari *proporsional* dihitung dengan mengalikan nilai *error* dengan konstanta K_p . Berikut rumus *Proporsional*.

$$P_{Out} = K_p e(t) \dots\dots\dots(2.3)$$

$$P_{Out} = K_p (Y_{sp} - Y_m) \dots\dots\dots(2.4)$$

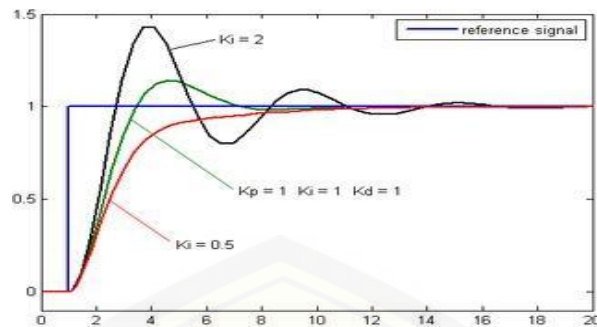
Konstanta K_p dapat disebut dengan *gain proporsional* atau *gain* kontroler, yang sangat berpengaruh pada output alat atau *system*. Jika K_p besar maka menghasilkan perubahan besar, dan mengakibatkan *system* tidak stabil, sedangkan jika konstanta K_p terlalu kecil akan berakibat respon pada *system* kecil pula. Sehingga *system* menjadi kurang *responsive* terhadap gangguan atau *error*. Berikut gambar 2.11 ilustrasi pengaruh K_p pada respon PID.



Gambar 2. 11 Pengaruh Konstanta K_p Terhadap Respon Kontrol PID
(Fitrianda, 2013a)

Integral disini berfungsi untuk mempercepat *system* menuju *set point* tertentu, akan tetapi *integral* disini akan menghilangkan *steady state error* yang muncul pada kontrol *proporsional*. Kontrol ini mempunyai kekurangan yang dapat menyebabkan terjadinya *overshoot* pada respon *system*, karena *integral* sendiri merespon *error* yang terakumulasi dari sebelumnya. Keluaran didapat dari mengakumulasi *error* yang kemudian dikalikan dengan konstanta *integral*. Berikut rumus pada *integral* PID dan Gambar 2.12 merupakan ilustrasi pengaruh K_i terhadap respon PID.

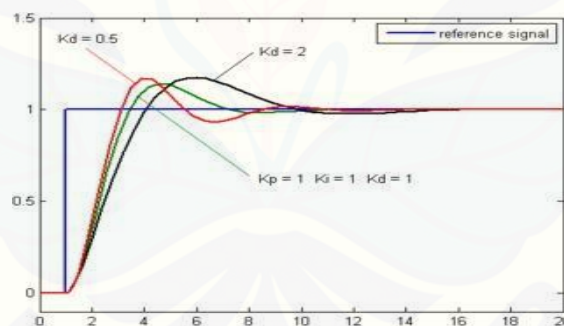
$$I_{Out} = K_i \int_0^t e(\tau) d\tau \dots\dots\dots(2.5)$$



Gambar 2. 12 Pengaruh konstanta K_i terhadap respon kontrol PID
(Fitrianda, 2013a)

Derivative berfungsi sebagai prediksi perilaku *system* yang kemudian memperbaiki waktu tinggal dan stabilitas *system*. Mendapatkan keluaran dari *system* itu yaitu dengan proses turunan *error* pada proses dihitung dengan menentukan kemiringan *error* pada setiap waktu dan mengalikan perubahan tiap waktu dengan konstanta K_d , berikut rumus *Derivative* pada PID dan Gambar 2.13 merupakan ilustrasi pengaruh K_d terhadap response PID pada *system*.

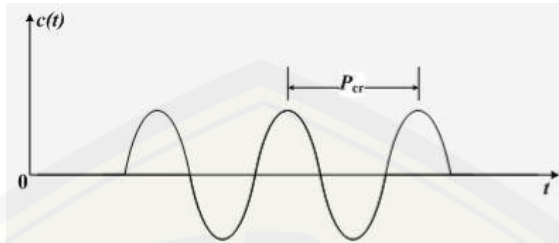
$$D_{out} = K_d \frac{de(t)}{dt} \dots\dots\dots(2.6)$$



Gambar 2. 13 pengaruh K_d terhadap response PID
(Fitrianda, 2013a)

Jenis *Tunning* pada PID Antara lain menggunakan Metode *Tunning Zieger-Nichols*. Dimana dalam metode kedua *Ziegler-Nichols*, mula-mula yang dilakukan adalah membuat $T_i = 0$ dan $T_d = 0$. Kemudian hanya dengan menggunakan tindakan kontrol *proporsional*, harga ditingkatkan dari nol ke suatu nilai kritis K_{cr} , disini mula-mula keluaran memiliki *osilasi* yang berkesinambungan. Dari keluaran yang berosilasi secara berkesinambungan, penguatan kritis K_{cr} dan periode P_{cr} dapat ditentukan. Untuk osilasi

berkesinambungan dengan periode P_{cr} ditunjukkan dalam Gambar 2.14 *Ziegler-Nichols* menyarankan penyetelan nilai parameter K_p , K_i , K_d dan berdasarkan rumus yang ditunjukkan dalam Tabel 2.2 (Ogata, K., 1997)



Gambar 2. 14 Osilasi berkesinambungan dengan periode P_{cr}

(Ogata, K., 1997)

Tabel 2. 2 Aturan Dasar *Ziegler-Nichols* Berdasarkan K_{cr} dan P_{cr}

| Tipe Kontroler | K_p | T_i | T_d |
|-----------------------|-------------------------|-------------------------|-------------------------|
| P | $0.5 K_{cr}$ | ∞ | 0 |
| PI | $0.45 K_{cr}$ | $\frac{1}{1.2} P_{cr}$ | 0 |
| PID | $0.60 K_{cr}$ | $0.5 P_{cr}$ | $0.125 P_{cr}$ |

BAB 3 METODE PELAKSANAAN DATA

3.1 Tempat dan Waktu Penelitian

Tempat penelitian bertempat di Laboratorium Sistem Cerdas dan Robotika, Gedung CDAST Lantai 7, Universitas Jember, dan bertempat di Laboratorium Elektronika dan Terapan Fakultas Teknik, Jl Slamet Riyadi, Patrang, Jember, serta bertempat di Lembaga Pengabdian Masyarakat, Universitas Jember Jl. Veteran, Jember. Waktu penelitian ini dimulai pada bulan November 2019 – Mei 2020, dengan rincian seperti pada tabel 3.1 Rencana dan Jadwal Pelaksanaan Penelitian sebagai berikut :

Tabel 3. 1 Rencana dan Jadwal Pelaksanaan Penelitian

| No | Kegiatan | Bulan | | | | | |
|----|-------------------|-------|----|-----|----|---|----|
| | | I | II | III | IV | V | VI |
| 1 | Studi Literatur | ■ | | | | | |
| 2 | Perancangan | | ■ | ■ | ■ | ■ | |
| 3 | Pengambilan Data | | | ■ | ■ | ■ | |
| 4 | Analisa Data | | | | ■ | ■ | |
| 5 | Pembuatan Laporan | | | | ■ | ■ | ■ |

3.2 Tahapan Penelitian

Langkah-langkah yang akan dilakukan dalam penelitian ini adalah sebagai berikut

1. Studi Literatur

Tahap awal pelaksanaan penelitian ini adalah dengan mencari literatur dari hasil penelitian sebelumnya melalui buku atau internet untuk

mengetahui karakteristik komponen sistem, prinsip kerja serta teori yang menunjang lainnya. Diharapkan dengan literatur yang telah didapat dapat memberikan arahan untuk mengurangi kesalahan dalam penelitian.

2. Perancangan Alat

Tahapan ini merupakan tahap merancang konstruksi secara sistematis dari alat yang akan dilakukan penelitian. Diharapkan dari proses perancangan konstruksi yang sistematis ini, alat yang nantinya akan diteliti dapat terbentuk. Hal – hal lain yang dilakukan yakni seperti penyoderan, perancangan sistem monitoring.

3. Implementasi Alat

Setelah *hardware* dan *software* terbentuk maka dilakukan pengujian pada masing-masing blok dan kemudian pengujian dilakukan pada keseluruhan sistem. Dalam implementasi alat ini juga dilakukan proses kalibrasi dimana pada proses kalibrasi ini bertujuan agar pembacaan sensor akurat sesuai dengan alat ukur.

4. Analisa dan Pengambilan Data

Setelah melakukan pengujian pada keseluruhan sistem dan memastikan bekerja dengan baik dan hasilnya memenuhi target, maka yang dilakukan selanjutnya adalah pengambilan data yang diperlukan untuk kemudian dianalisa dari data yang telah didapatkan.

5. Penyusunan Laporan

Pada tahap akhir ini, hasil pengambilan data dan analisa dimasukkan ke pembahasan. Kemudian, dari apa yang telah dianalisa dapat ditarik beberapa kesimpulan yang menyangkut kinerja dari alat yang dibuat dan memberikan saran untuk memperbaiki kekurangan yang ada, kemungkinan pengembangan, serta penyempurnaan alat di masa mendatang.

3.3 Alat dan Bahan

Peralatan – peralatan yang diperlukan dalam penelitian adalah :

1. *Personal Computer* (PC)
2. Alat – alat ukur : *Multimeter, DC Power Supply*

3. Perkakas mekanis : Palu, Solder, Tang, Obeng, Gergaji, Gerinda, Bor, Penggaris Sudut (Busur), *Tachometer*
4. Perangkat lunak: *Windows 7, Arduino IDE.*

Beberapa komponen yang diperlukan dalam penelitian ini adalah :

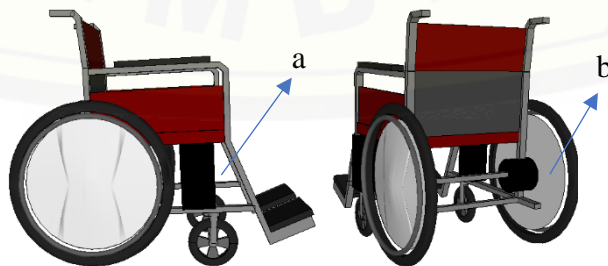
1. *Arduino Nano*
2. *Rotary Encoder*
3. *Sensor Gyroscope*
4. 1 set Kursi Roda
5. *STM32F103 (Bluepill)*
6. Baterai 12 V
7. *nRF 24L01*
8. PCB polos

3.4 Perancangan mekanik Kursi Roda dan Topi Kendali

Perancangan mekanik pada kursi roda terdiri dari dua bagian yang pertama adalah bagian kursi roda itu sendiri yang terdapat pada Gambar 3.1 Kursi Roda dan yang kedua yaitu bagian kendali pada kursi roda berupa topi yang terdapat pada Gambar 3.2 Topi Kendali.

3.4.1 Perancangan Mekanik Kursi Roda

Pada kursi roda menggunakan kursi roda konvensional pada umumnya yang telah di modifikasi. Kursi roda konvensional tersebut hanya di ambil rangka dan motornya berserta battery, dan menghilangkan bagian kontrol (*joystick*) pada kursi roda. Berikut gambar 3.1 *design* kursi roda yang akan di pakai pada penelitian ini



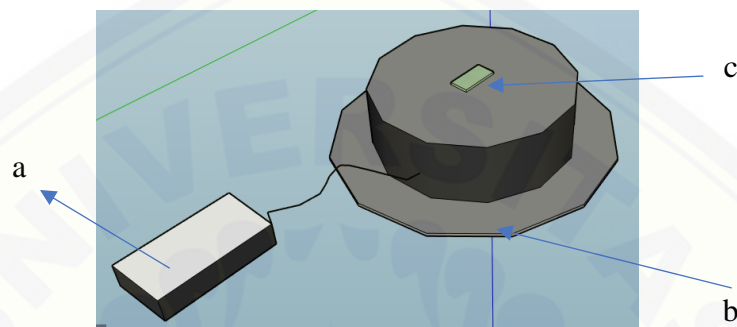
Gambar 3. 1 *Design* Kursi Roda

Keterangan :

- a. Box komponen
- b. Motor DC

3.4.2 Perancangan Mekanik Topi Kendali

Pada perancangan mekanik kendali sendiri Untuk kendalinya menggunakan topi bundar yang telah di modifikasi untuk letak modul sensor dan komponen kendali. Pada topi hanya terdapat modul sensor sedangkan untuk perangkat elektronik yang lain berada di box terpisah seperti pada gambar 3.2 design topi kendali di bawah ini.



Keterangan :

Gambar 3. 2 *Design Topi Kendali*

- a. Box komponen
- b. Topi
- c. Sensor *Gyroscope*

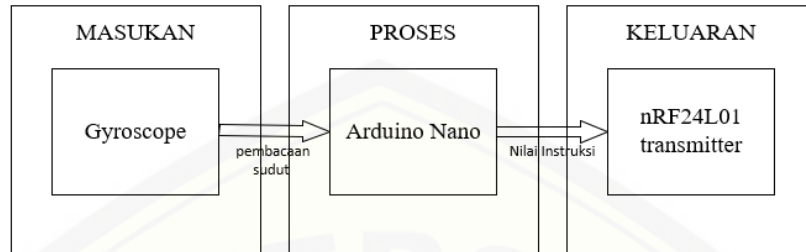
3.5 Perancangan Sistem Elektronika Kursi Roda dan Topi Kendali

Untuk perancangan elektronika pada penelitian ini juga terdapat dua bagian, yaitu bagian elektrikal pada kursi roda dan bagian elektrikal pada kendali. Untuk bagian kursi roda terdapat catu daya, *driver* motor, Motor DC, modul *receiver* dan mikrokontroler, sedangkan untuk elektrikal kendali terdapat catu daya mikrokontroler, sensor *gyroscope* dan modul *transmitter* untuk mengirim data sensor ke komponen kursi roda.

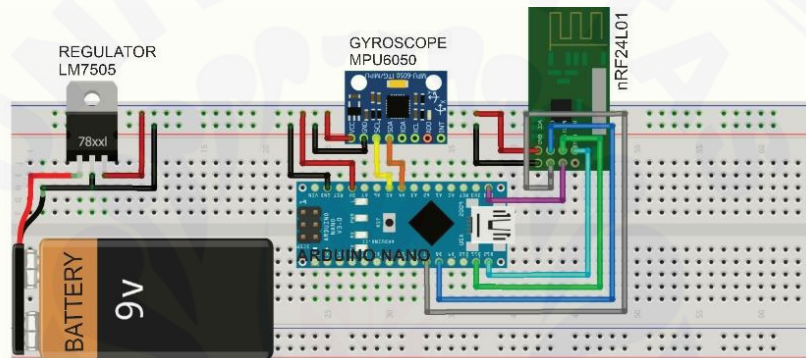
3.5.1 Perancangan Sistem Elektronika Topi Kendali.

Dalam perancangan sistem kendali kursi roda di butuhkan beberapa komponen berupa sensor *gyroscope*, modul nRF 24L01, mikrokontroler berupa arduino nano, dan battery. Pada sensor *gyroscope* digunakan untuk mendeteksi adanya kemiringan pada kepala yang mana digunakan sebagai instruksi arah robot untuk bejalan, dan untuk mikrokontroller sendiri digunakan sebagai pemrosesan hasil pembacaan sensor, sedangkan untuk modul nRF 24L01 digunakan mengirim hasil pembacaan sensor *gyroscope* pada kontroler kursi roda. Pada gambar 3.3

Diagram Blok Sistem Topi Kendali merupakan gambar diagram blok pada kendali kusi roda, dan pada Gambar 3.4 Sistem Elektronika Topi Kendali merupakan gambar pengkabelan pada kendali kursi roda.

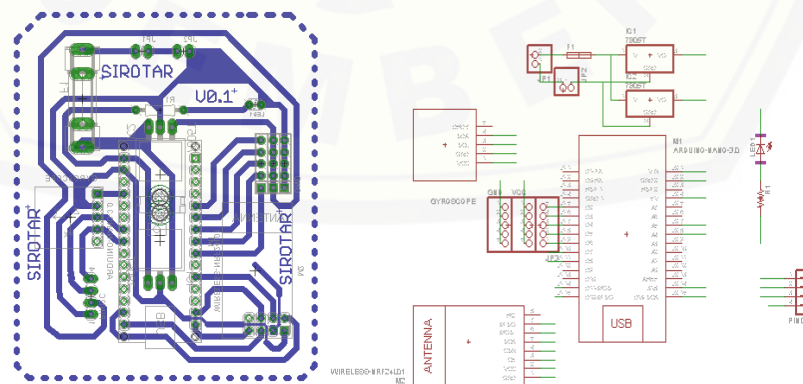


Gambar 3. 3 Diagram Blok sistem elektronika Kendali



Gambar 3. 4 Sistem Elektronika Topi Kendali

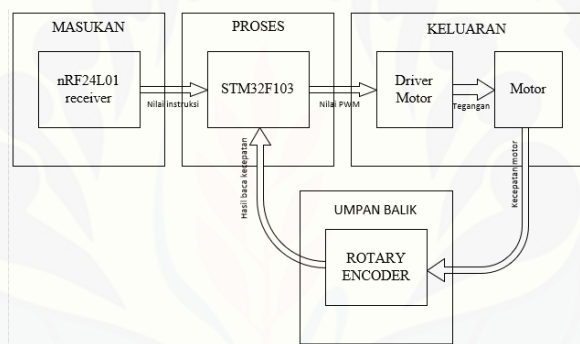
Berikut dibawah ini berupa *design schematic* dan *layout* pada *shield* elektronika Topi Kendali yang di design menggunakan *software design pcb* pada penelitian ini. Berikut gambar 3.5 (a) Layout Elektronika Topi Kendali dan (b) Schematic Elektronika Topi Kendali.



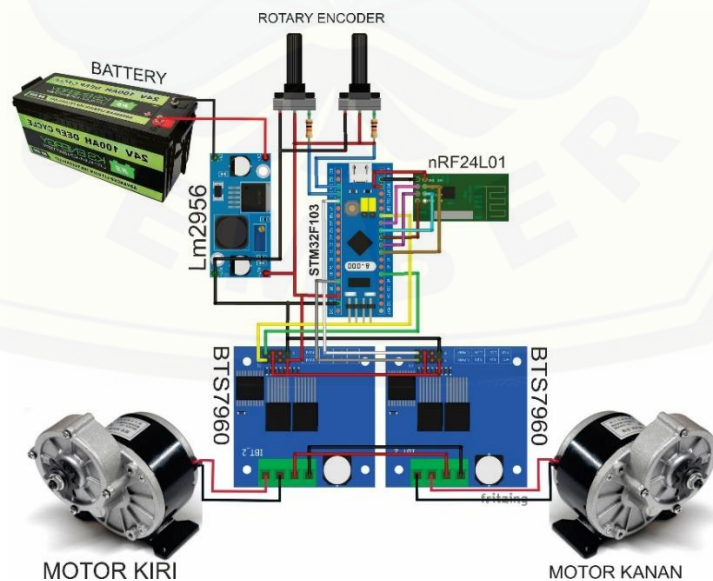
Gambar 3. 5 (a) Layout Elektronika Topi Kendali (b) Schematic Elektronika Topi Kendali.

3.5.2 Perancangan Sistem Elektronika Kursi Roda

Pada perancangan elektronika kursi roda terdapat beberapa modul dan komponen yang akan digunakan yaitu modul nrf 24L01 yang digunakan untuk menerima data dari pembacaan sensor kendali yang telah di proses, kemudian mikrokontroler yang digunakan untuk memproses hasil penerimaan data yang kemudian menginstruksikan *driver* motor untuk mengedalikan putaran motor pada kursi roda, sedangkan pada rotari sendiri digunakan untuk mengatur kecepatan pada putaran roda kursi roda. Pada Gambar 3.6 Diagram Blok Sistem Elektronika Kursi Roda merupakan gambar diagram blok pada sistem elektronika kursi roda, dan pada Gambar 3.7 Sistem Elektronika Kursi Roda merupakan gambar pengkabelan pada sistem elektronika kursi roda.

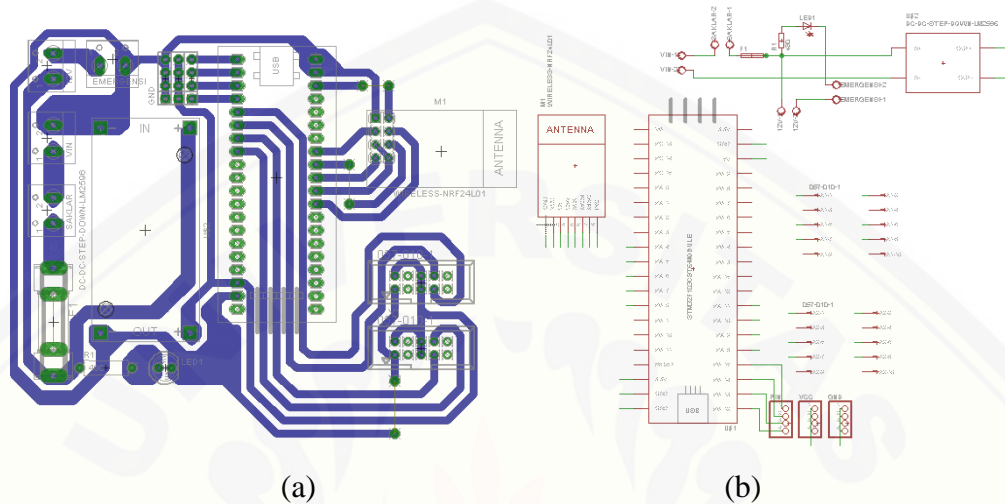


Gambar 3. 6 Diagram Blok Sistem Elektronika Kursi Roda



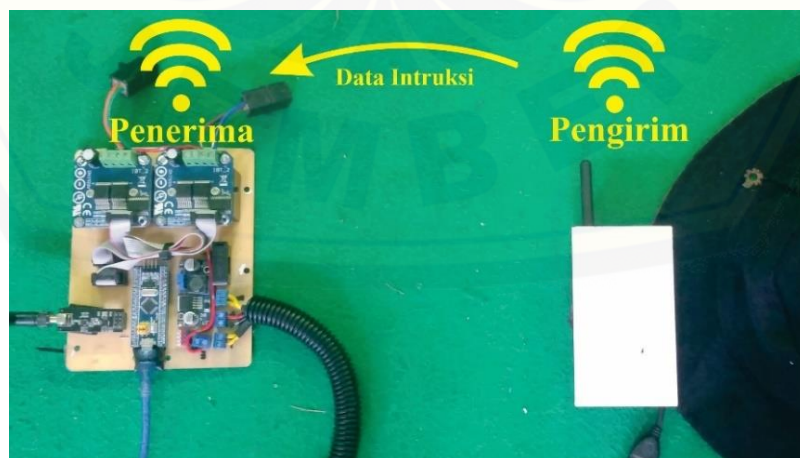
Gambar 3. 7 Sistem Elektronika Kursi Roda

Berikut dibawah ini berupa *design schematic* dan *layout* pada *shield* elektronika kursi roda yang di design menggunakan *software design pcb* pada penelitian ini. Berikut gambar 3.8 (a) *Layout* Elektronika Kursi Roda dan (b) *Schematic* Elektronika Kursi Roda.



Gambar 3. 8 (a) *Layout* elektronika Kursi Roda (b) *Schematic* Kursi Roda

Kedua rangkaian diatas antara rangkaian topi kendali dan rangkaian kursi roda nantinya akan saling berkomunikasi. Topi kendali nantinya akan mengirim beberapa data instruksi yang nantinya akan di proses oleh rangkaian pada kursi roda berikut gambar ilustrasinya pada gambar 3.9 Kinerja rangkaian topi kendalul dan rangkaian kursi roda.



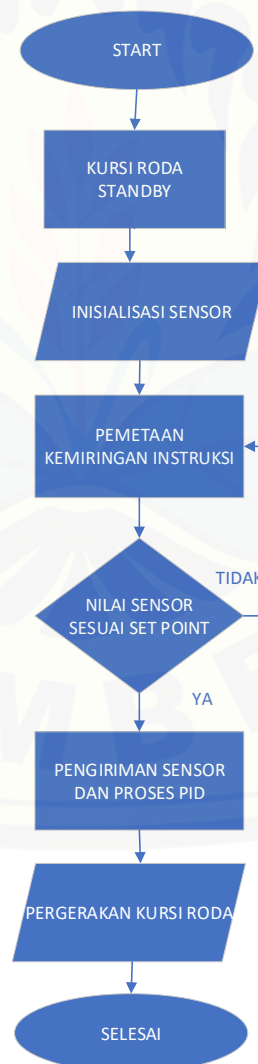
Gambar 3. 9 Kinerja rangkaian topi kendali dan rangkaian kursi roda

3.6 Perancangan Perangkat Lunak

Perancangan perangkat lunak sendiri menggambarkan proses alur pemrograman dan bagaimana suatu *system* dapat bekerja serta terprogram sesuai yang di kehendaki atau yang telah di tentukan alur pergerakannya.

3.6.1 Perancangan Perangkat Lunak Secara Keseluruhan.

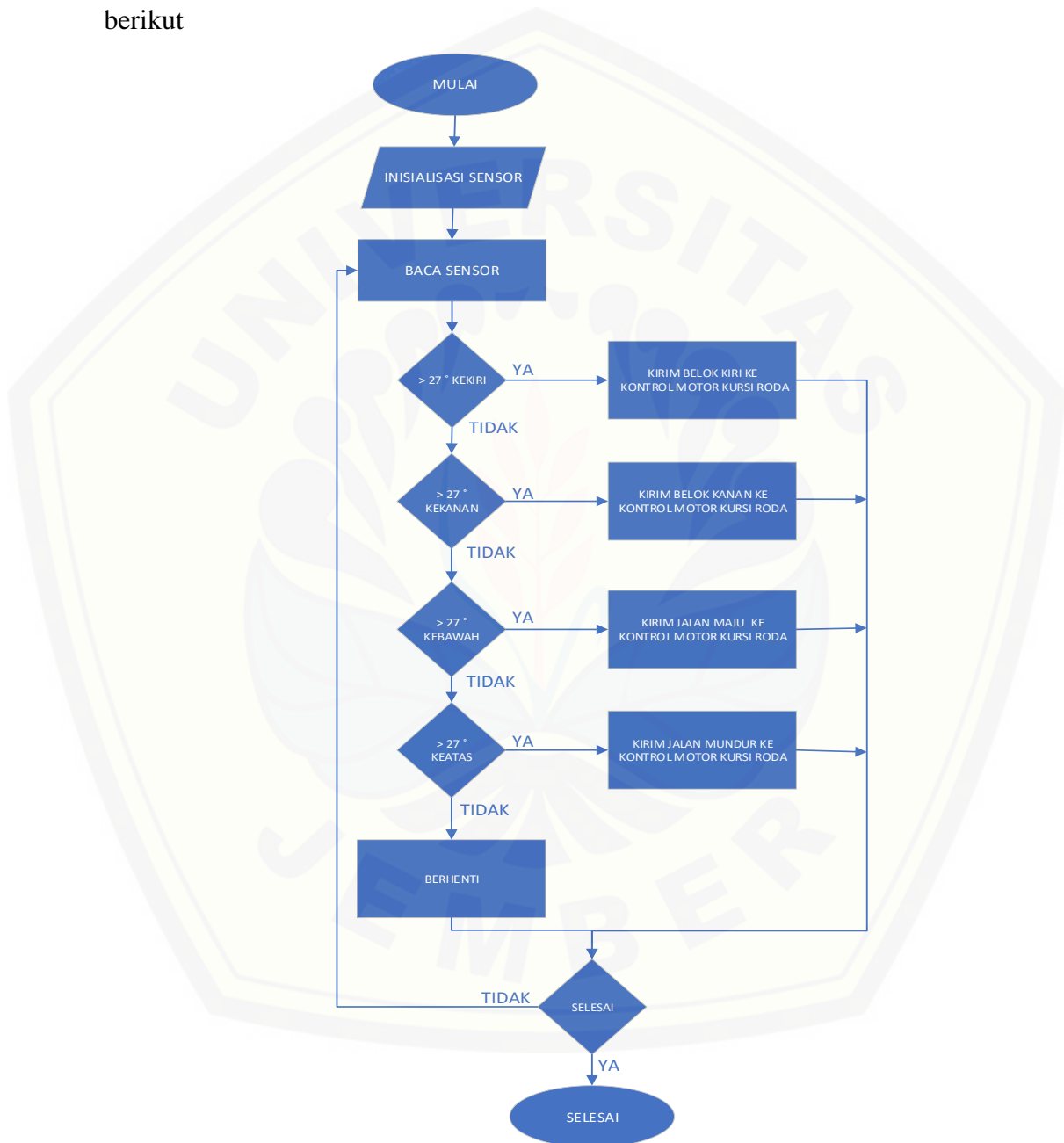
Perancangan perangkat lunak nantinya akan menggunakan *software arduino* untuk memprogram alat baik dari sistem kendalinya maupun pada kinerja kursi roda itu sendiri. Berikut Gambar 3.10 Diagram Alir Alat yang merupakan gambaran algoritma pada kerja alat secara keseluruhan.



Gambar 3. 10 Diagram Alir Alat Keseluruhan

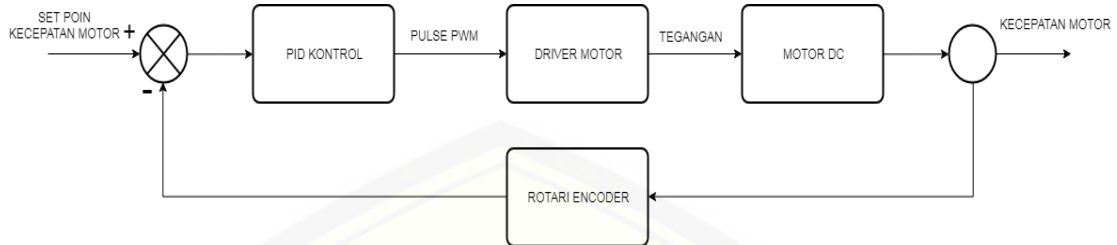
3.6.2 Perancangan Perangkat Lunak Pada Topi Kendali.

Didalam pemrosesan diagram alur pada perancangan perangkat lunak secara keseluruhan terdapat pula diagram alur pemrosesan topi kendali pada penelitian seperti pada gambar 3.11 diagram alir alat bagian topi kendali sebagai berikut



Gambar 3. 11 Diagram Alir Alat Bagian Topi Kendali

3.6.3 Perancangan Kontrol dan Tuning PID

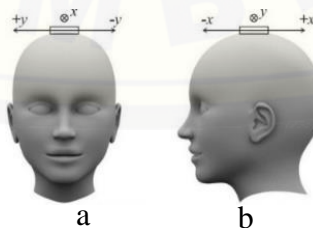


Gambar 3. 12 Diagram blok kontrol PID Motor Kursi Roda

Pada proses *tuning* kontrol PID dengan menggunakan metode osilasi *Ziegler-Nichols* dimulai dengan memberikan nilai 0 pada parameter T_i dan T_d . Sedangkan nilai K_p dinaikkan sedikit demi sedikit hingga didapatkan grafik posisi Kecepatan yang berkesinambungan. Kesinambungan yang dimaksud adalah saat grafik memiliki amplitudo yang sama pada setiap periodenya. Setelah didapatkan grafik yang berkesinambungan langkah selanjutnya adalah menghitung nilai K_{cr} dan P_{cr} . K_{cr} adalah nilai K_p saat terjadi osilasi berkesinambungan sedangkan P_{cr} adalah periode kesinambungan dari grafik. Setelah didapatkan K_{cr} dan P_{cr} langkah selanjutnya adalah menghitung nilai K_p , T_i dan T_d sesuai dengan aturan dari *Ziegler-Nichols*. Gambar 3.12 Diagram blok kontrol PID motor kursi roda diatas merupakan gambar dari desing kontrol PID yang di gunakan.

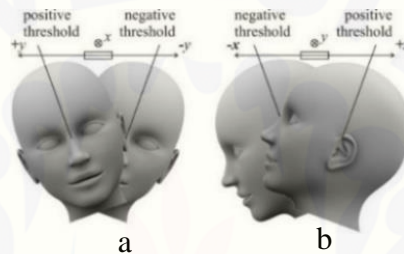
3.7 Pola Gerak Kendali Pada Kepala

Pada pola gerak ini menjelaskan bagaimana kendali kursi roda menggunakan gerakan kepala, dimana pada gambar 3.13 Posisi awal kepala, menjelaskan bahwa tidak ada instruksi pergerakan pada kursi roda.



Gambar 3. 13 Posisi Awal Kepala (a) Tampak Depan (b) Tampak Samping
(Pajkanovic dan Dokic, 2013)

Sedangkan pada Gambar 3.14 Pola Pergerakan menjelaskan bagaimana kendali pada kursi roda saat kursi roda akan bergerak maju dan mundur maupun belok kanan dan kiri. Untuk intruksi maju sendiri tetap seperti pada gambar 3.14 pola pergerakan (b) maju- mundur hanya saja untuk instruksi maju hanya satu anggukan untuk intruksi maju dan ketika ingin berhenti dari instruksi maju hanya cukup dengan menganggukan kepala satu kali lagi, sedangkan untuk instruksi belok kanan, belok kiri dan mundur, ketika belok kiri harus memiringkan kepala kekiri, belok kanan memiringkan ke kanan dan mundur dengan mendongakkan kepala ke atas sampai kursi roda berada di kondisi yang di tentukan.



Gambar 3. 14 Pola Pergerakan (a) Belok Kanan-Kiri, dan (b) Maju-Mundur
(Pajkanovic dan Dokic, 2013)

3.8 Design Pengambilan dan pengujian Data

Pengambilan data dilakukan agar mengetahui tingkat kelayakan komponen yang digunakan dan untuk mengetahui hasil penelitian yang telah dilakukan.

3.8.1 Data Transfer Data nRF24L01

Pengujian Transfer data dilakukan dengan menguji modul nRF 24L01 seberapa jauh jarak kendali dapat mengirim sinyal ke kursi roda dan bagaimana proses pengiriman data padatopi kendali ke kursi roda.

3.8.2 Data Gyroscope

Pada pengujian *Gyroscope* yaitu menguji nilai kemiringan sudut pada modul yang digunakan. Dimana pada pengujian ini akan membandingkan nilai yang dibaca oleh modul *gyroscope* pada alat dengan penggaris sudut atau busur.,

3.8.3 Pengujian Motor

Pada pengujian tahap ini yaitu pengujian terhadap aktuator penggerak kursi roda berupa motor dc menggunakan perangkat berupa *Tachometer*. Dimana pada pengujian ini nantinya membandingkan nilai kecepatan yang dibaca oleh sensor *Rotary* dengan *Tachometer*.

3.8.4 Pengujian kontrol PID

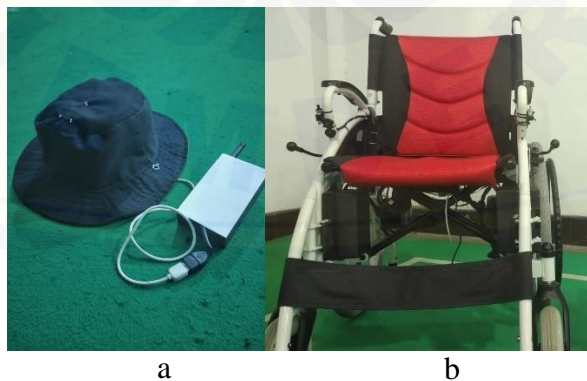
Pada pengujian kontrol PID diharapkan motor mampu bergerak sesuai dengan respon yang telah ditentukan. Untuk itu dibutuhkan proses *tunning* pada parameter PID. Pada pengujian ini dilakukan pengujian PID dengan beban pada kursi roda dengan parameter yang telah di tentukan.

3.8.5 Pengujian Keseluruhan

Pada pengujian Keseluruhan, alat nantinya akan di uji dengan harapan kursi roda mampu bergerak menggunakan kendali berupa sensor *gyroscope* dan pergerakan kursi roda dengan kontrol PID pada Motor.

3.8.6 Hasil Perancangan Alat

Hasil perancangan alat merupakan pemaparan dari hasil pembuatan yang telah di selesaikan berdasarkan *design* yang telah di buat sebelumnya. Hasil perancangan alat disini berupa topi kendali dan bagian bagian kursi roda yang telah di susun. Berikut gambar 3.15 Topi Kendali dan Kursi Roda



Gambar 3. 15 (a) Kursi Roda (b) kendali kursi roda

BAB 5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil yang telah diperoleh dalam penelitian yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut:

1. Pada topi kendali menggunakan *range* kemiringan 27 derajat untuk menjalankan intruksi, yaitu kemiringan lebih dari 27 derajat ke kanan maka instruksi belok kanan, kemiringan 27 derajat ke kiri maka instruksi belok ke kiri, kemiringan 27 derajat ke bawah instruksi gerak maju, dan kemiringan ke atas maka instruksi gerak mundur. Hasil tersebut dapat dilihat pada gambar 4.8 sampai dengan 4.12 pada gambar pengiriman data
2. Pada proses *tunning* PID didapat hasil yang lebih bagus dari *tunning* PID *tunning trial and error* yaitu memiliki *overshoot* (Mp%) sebesar 7.1%, *Peak Time* sebesar 0.3 s, serta *Rise Time* sebesar 1 s. Sedangkan *tunning* PID menggunakan *Ziegler Nichols* memiliki *overshoot* (Mp%) sebesar 94.9%, *Peak Time* sebesar 0.4 s, serta *Rise Time* sebesar lebih dari 5.4 s. Dari perbedaan tersebut *tunning* PID dengan *trial error* memiliki kelebihan Respon lebih stabil, dan cepat mencapai *steady state* dibandingkan metode *Ziegler-Nichols*. Hasil tersebut dapat dilihat pada tabel 4.11 Perbandingan kontrol PID *tunning* manual dan *Ziegler Nichols*.
3. Pada pengujian bersama pengguna kursi roda, pengguna harus menyesuaikan diri dengan pergerakan kursi roda agar pengguna mengetahui karakteristik pergerakan pada kursi roda.

5.2 Saran

Dengan harapan penelitian ini dapat dikembangkan dan disempurnakan dikemudian hari, penulis memberikan saran sebagai berikut:

1. Untuk pergerakan dan akselerasi pada kursi roda lebih di perhalus dengan menggunakan metode yang lain. Seperti kemiringan sudut pada instruksi *proporsional* dengan kecepatan.
2. Lebih di tingkatkan untuk masalah keselamatan pengguna kursi roda, seperti tombol darurat pada kursi roda.

DAFTAR PUSTAKA

- Batan, I. M. L. 2006. Pengembangan kursi roda sebagai upaya peningkatan ruang gerak penderita cacat kaki. *Jurnal Teknik Industri*. 8(2):97–105.
- Fitrianda, M. I. 2013a. *Digital Digital Repository Repository Universitas Universitas Jember Jember Digital Digital Repository Repository Universitas Universitas Jember*
- Fitrianda, M. I. 2013b. Digital digital repository repository universitas universitas jember jember digital digital repository repository universitas universitas jember
- Hoch, S., H. Kunreuther, D. A. Torres, L. D., Aguilar, J., Hernandez, D. A. & Vera, S. Licenciada, E. Socioculturales title. *Journal of Linguistics*. 3(2):139–157.
- Pajkanovic, A. dan B. Dokic. 2013. Wheelchair control by head motion. *Serbian Journal of Electrical Engineering*. 10(1):135–151.
- Pradito, D. 2010. Prinsip kerja dan sistem gerak roda omni pada robot badminton. 5–28.
- Sadewo, A. D. B., E. R. Widasari, dan A. Muttaqin. 2017. Perancangan pengendali rumah menggunakan smartphone android dengan konektivitas bluetooth. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*. 1(5):415–425.
- Simpson, R., E. LoPresti, S. Hayashi, I. Nourbakhsh, dan D. Miller. 2004. The smart wheelchair component system. *Journal of Rehabilitation Research and Development*. 41(3 B):429–442.

LAMPIRAN

A. Program STM32F103

A.1 Program utama

```

/*.....nrf.....*/
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
boolean state0, state1, state2, state3;
RF24 radio(PB0, PA4); // CE, CSN on Blue Pill
const uint64_t address = 0xF0F0F0F0E1LL;
int kirimz;
/*-----Rotary-----*/
double rpskiri, rpmkiri, rpskanan, rpmkanan;
int  posisikiri,  lastjumlahPulsakiri,  posisikanan,
lastjumlahPulsakanan;
int intervalkiri = 100;
int intervalkanan = 100;
unsigned  int  waktulamakiri,  waktusekarangkiri,
waktulamakanan, waktusekarangkanan;
/*-----INIALISASIPIDKIRI-----
*/
float  pid_outputkiri,  pid_i_memkiri,  pid_errorkiri,
pid_last_errorkiri;
float targetkiri = 40;
float pid_maxkiri = 100;
float pid_pkiri = 0.011  ,  pid_ikiri  = 0.00011,
pid_dkiri = 7;
/*-----INIALISASIPIDKANAN-----
-*/

```

```
float pid_outputkanan, pid_i_memkanan, pid_errorkanan,
pid_last_errorkanan;
float targetkanan = 40;
float pid_maxkanan = 100;
float pid_pkanan = 0.011 , pid_ikanan = 0.00011,
pid_dkanan = 7;
/*.....motor.....*/
int motorkiri1 = PA8; // motorkiri
int motorkiri2 = PB8; // motorkiri
int motorkanan1 = PA1; // motorkanan
int motorkanan2 = PB1; // motorkanan
int motorkiri1pwm = 0; // motorkiri
int motorkiri2pwm = 0; // motorkiri
int motorkanan1pwm = 0; // motorkanan
int motorkanan2pwm = 0; // motorkanan
int pwmtarget = 100;
int count;
int count2;
int trigger;
void setup() {
  /*.....nrf.....*/
  radio.begin();
  radio.openReadingPipe(0, address); //Setting the
address at which we will receive the data
  radio.setPALevel(RF24_PA_MIN); //You can set
this as minimum or maximum depending on the distance
between the transmitter and receiver.
  radio.startListening(); //This sets the
module as receiver
  /*.....motor.....*/
  Serial.begin(115200);
```

```
pinMode(motorkiri1, OUTPUT); // set pin
mode motorkiri1 as PWM output
pinMode(motorkiri2, OUTPUT); // set pin
mode motorkiri2 as PWM output
pinMode(motorkanan1, OUTPUT); // set pin
mode motorkanan1 as PWM output
pinMode(motorkanan2, OUTPUT); // set pin
mode motorkanan2 as PWM output
Serial.begin(9600);
attachInterrupt(digitalPinToInterrupt(PB15),
hitung_posisikiri, FALLING);
attachInterrupt(digitalPinToInterrupt(PB14),
hitung_posisikanan, FALLING);
}
void loop() {
awal:
pid();
// put your main code here, to run repeatedly:
//char text[32] = ""; //Saving the incoming
data
radio.read(& kirimz, sizeof(kirimz)); //Reading the
data
delay(50);
Serial.print(count);
Serial.println(kirimz);
if (trigger == 1) {
while (1) {
motorkiri2pwm = motorkiri2pwm + 10;
motorkanan2pwm = motorkanan2pwm + 10;
if (motorkiri2pwm > pwmtarget) {
motorkiri2pwm = pwmtarget;
```

```
    }
    if (motorkanan2pwm > pwmtarget) {
        motorkanan2pwm = pwmtarget;
    }
    maju();
    radio.read(& kirimz, sizeof(kirimz)); //Reading
the data
    delay(50);
    break;
}
if (count2 == 0) {
    if (kirimz == 0) {
        count2++;
    }
}
else if (count2 == 1) {
    if (kirimz == 3) {
        count2++;
    }
}
else if (count2 == 2) {
    motorkirilpwm = 0;
    motorkiri2pwm = 0;
    motorkanan1pwm = 0;
    motorkanan2pwm = 0;

    mandek();
    if (kirimz == 0){
        trigger = 0;
    }
}
```

```
}  
else if (trigger == 0) {  
    count2 = 0;  
    if ( kirimz == 0) {  
        motorkirilpwm = 0;  
        motorkiri2pwm = 0;  
        motorkanan1pwm = 0;  
        motorkanan2pwm = 0;  
        mandek();  
    }  
    else if ( kirimz == 3)  
    {  
        trigger = 1;  
    }  
}  
if ( kirimz == 1)  
{  
    trigger = 0;  
    while (1) {  
        motorkirilpwm = motorkirilpwm + 10;  
        motorkanan2pwm = motorkanan2pwm + 10;  
        if (motorkirilpwm > pwmtarget) {  
            motorkirilpwm = pwmtarget;  
        }  
        if (motorkanan2pwm > pwmtarget) {  
            motorkanan2pwm = pwmtarget;  
        }  
        kanan();  
        radio.read(&kirimz, sizeof(kirimz));    //Reading  
the data  
        delay(50);  
    }  
}
```

```
        break;
    }
}
else if (kirimz == 2)
{
    trigger = 0;
    while (1) {
        motorkiri2pwm = motorkiri2pwm + 10;
        motorkanan1pwm = motorkanan1pwm + 10;
        if (motorkiri2pwm > pwmtarget) {
            motorkiri2pwm = pwmtarget;
        }
        if (motorkanan1pwm > pwmtarget) {
            motorkanan1pwm = pwmtarget;
        }
        kiri();
        radio.read(&kirimz, sizeof(kirimz)); //Reading
the data
        delay(50);
        break;
    }
}
else if (kirimz == 4)
{
    trigger = 0;
    while (1) {
        motorkirilpwm = motorkirilpwm + 10;
        motorkanan1pwm = motorkanan1pwm + 10;
        if (motorkirilpwm > pwmtarget) {
            motorkirilpwm = pwmtarget;
        }
    }
}
```

```
        if (motorkanan1pwm > pwmtarget) {
            motorkanan1pwm = pwmtarget;
        }
        mundur();
        radio.read(& kirimz, sizeof(kirimz)); //Reading
the data
        delay(50);
        break;
    }
}
```

```
void hitung_posisikiri () {
    posisikiri++; // rutin interrupt 0 (INT 0)
}
void hitung_posisikanan () {
    posisikanan++; // rutin interrupt 0 (INT 0)
}
```

A. 2 Program Motor

```
void maju()
{
    analogWrite(motorkiri1,0);
    analogWrite(motorkanan1, 0);
    analogWrite(motorkiri2,motorkiri2pwm);
    analogWrite(motorkanan2, motorkanan2pwm);
}
```

```
void kiri()
{
    analogWrite(motorkiri1,0);
    analogWrite(motorkanan1, motorkanan1pwm);
}
```

```
analogWrite (motorkiri2,motorkiri2pwm);
analogWrite (motorkanan2, 0);
}
void kanan()
{
analogWrite (motorkiril,motorkirilpwm);
analogWrite (motorkanan1, 0);
analogWrite (motorkiri2,0);
analogWrite (motorkanan2, motorkanan2pwm);
}
void mundur()
{
analogWrite (motorkiril,motorkirilpwm);
analogWrite (motorkanan1, motorkanan1pwm);
  nalogWrite (motorkiri2,0);
analogWrite (motorkanan2,0);
}
void mandek()
{
analogWrite (motorkiril,0);
analogWrite (motorkanan1,0);
analogWrite (motorkiri2,0);
analogWrite (motorkanan2,0);
}
```

A. 3 Program PID

```
void pid()
{
  /*.....rotary.....*/
  waktusekarangkiri = millis();
  if (waktusekarangkiri - waktulamakiri >=
intervalkiri) {
```



```
rpskiri = ((posisikiri/7.75) / 360)*10 ;
rpmkiri = (rpskiri* 60);
//Serial.print(rpmkiri);
//Serial.print(" , ");
posisikiri = 0;
waktulamakiri = waktusekarangkiri;
}

waktusekarangkanan = millis();
if (waktusekarangkanan - waktulamakanan >=
intervalkanan) {
    rpskanan = ((posisikanan/7.75) / 360)*10;
    rpmkanan = (rpskanan* 60);
    //Serial.println(rpmkanan);
    posisikanan = 0;
    waktulamakanan = waktusekarangkanan;
}
/*-----PIDKIRI-----*/
pid_errorkiri = targetkiri - rpmkiri;
pid_i_memkiri += pid_ikiri * pid_errorkiri;
if (pid_i_memkiri > pid_maxkiri)pid_i_memkiri =
pid_maxkiri;
else if (pid_i_memkiri < pid_maxkiri * -
1)pid_i_memkiri = pid_maxkiri * -1;
pid_outputkiri = pid_pkiri * pid_errorkiri +
pid_i_memkiri + pid_dkiri * (pid_errorkiri -
pid_last_errorkiri);
if (pid_outputkiri > pid_maxkiri)pid_outputkiri =
pid_maxkiri;
else if (pid_outputkiri < pid_maxkiri * -
1)pid_outputkiri = pid_maxkiri * -1;
```

```

pid_last_errorkiri = pid_errorkiri;
/*-----PIDKANAN-----*/
pid_errorkanan = targetkanan - rpmkanan;
pid_i_memkanan += pid_ikanan * pid_errorkanan;
if (pid_i_memkanan > pid_maxkanan)pid_i_memkanan =
pid_maxkanan;
else if (pid_i_memkanan < pid_maxkanan * -
1)pid_i_memkanan = pid_maxkanan * -1;
pid_outputkanan = pid_pkanan * pid_errorkanan +
pid_i_memkanan + pid_dkanan * (pid_errorkanan -
pid_last_errorkanan);
if (pid_outputkanan > pid_maxkanan)pid_outputkanan =
pid_maxkanan;
else if (pid_outputkanan < pid_maxkanan * -
1)pid_outputkanan = pid_maxkanan * -1;
pid_last_errorkanan = pid_errorkanan;
}

```

B. Program Arduino Nano

B. 1 Program utama :

```

#include <Wire.h>
#include <nRF24L01.h> //NRF24L01 library created by
TMRh20 https://github.com/TMRh20/RF24
#include <RF24.h>
#include <SPI.h>
#define SwitchPin 8 // Arcade switch is connected to
Pin 8 on NANO
int SentMessage[1] = {000}; // Used to store value
before being sent through the NRF24L01
RF24 radio(7, 8); // CE, CSN
const uint64_t address = 0xF0F0F0F0E1LL;

```

```
int kirimz;
//Gyro Variables
float elapsedTime, time, timePrev; //Variables for time
control
int gyro_error=0; //We use this variable to only
calculate once the gyro data error
float Gyr_rawX, Gyr_rawY, Gyr_rawZ; //Here we store the
raw data read
float Gyro_angle_x, Gyro_angle_y; //Here we store the
angle value obtained with Gyro data
float Gyro_raw_error_x, Gyro_raw_error_y; //Here we
store the initial gyro data error
//Acc Variables
int acc_error=0; //We use this variable to only
calculate once the Acc data error
float rad_to_deg = 180/3.141592654; //This value is for
pasing from radians to degrees values
float Acc_rawX, Acc_rawY, Acc_rawZ; //Here we store the
raw data read
float Acc_angle_x, Acc_angle_y; //Here we store the
angle value obtained with Acc data
float Acc_angle_error_x, Acc_angle_error_y; //Here we
store the initial Acc data error
float Total_angle_x, Total_angle_y;
// Serial output refresh time
const long SERIAL_REFRESH_TIME = 100;
long refresh_time;
float nilai_sudutx;
float nilai_suduty;
int trigger;
int count;
```

```
void setup() {
    radio.begin();//Starting the Wireless communication
    radio.openWritingPipe(address); //Setting the address
    where we will send the data
    radio.setPALevel(RF24_PA_MIN); //You can set it as
    minimum or maximum depending on the distance between
    the transmitter and receiver.
    radio.stopListening();//This sets the module as
    transmitter
    Wire.begin();//begin the wire communication
    Wire.beginTransmission(0x68); //begin, Send the slave
    adress (in this case 68)
    Wire.write(0x6B); //make the reset (place a 0 into
    the 6B register)
    Wire.write(0x00);
    Wire.endTransmission(true); //end the transmission
    //Gyro config
    Wire.beginTransmission(0x68); //begin, Send the slave
    adress (in this case 68)
    Wire.write(0x1B); //We want to write to the
    GYRO_CONFIG register (1B hex)
    Wire.write(0x10); //Set the register bits as 00010000
    (1000dps full scale)
    Wire.endTransmission(true); //End the transmission
    with the gyro
    //Acc config
    Wire.beginTransmission(0x68); //Start communication
    with the address found during search.
    Wire.write(0x1C); //We want to write to the
    ACCEL_CONFIG register
```

```

Wire.write(0x10); //Set the register bits as 00010000
(+/- 8g full scale range)
Wire.endTransmission(true);
Serial.begin(9600); //Remember to set this same baud
rate to the serial monitor
time = millis();//Start counting time in milliseconds
/*Here we calculate the acc data error before we start
the loop
* I make the mean of 200 values, that should be
enough*/
if(acc_error==0)
{
for(int a=0; a<200; a++)
{
Wire.beginTransaction(0x68);
Wire.write(0x3B); //Ask for the 0x3B register-
correspond to AcX
Wire.endTransmission(false);
Wire.requestFrom(0x68,6,true);
Acc_rawX=(Wire.read()<<8|Wire.read())/4096.0 ; //each
value needs two registres
Acc_rawY=(Wire.read()<<8|Wire.read())/4096.0 ;
Acc_rawZ=(Wire.read()<<8|Wire.read())/4096.0 ;
/*---X---*/
Acc_angle_error_x = Acc_angle_error_x +
((atan((Acc_rawY)/sqrt(pow((Acc_rawX),2) +
pow((Acc_rawZ),2))))*rad_to_deg));
/*---Y---*/
Acc_angle_error_y = Acc_angle_error_y + ((atan(-
1*(Acc_rawX)/sqrt(pow((Acc_rawY),2) +
pow((Acc_rawZ),2))))*rad_to_deg));

```

```
        if(a==199)
        {
            Acc_angle_error_x = Acc_angle_error_x/200;
            Acc_angle_error_y = Acc_angle_error_y/200;
            acc_error=1;
        }
    }
} //end of acc error calculation
/*Here we calculate the gyro data error before we start
the loop
* I make the mean of 200 values, that should be
enough*/
if(gyro_error==0)
{
    for(int i=0; i<200; i++)
    {
        Wire.beginTransmission(0x68); //begin, Send the
slave address (in this case 68)
        Wire.write(0x43); //First adress of the Gyro data
        Wire.endTransmission(false);
        Wire.requestFrom(0x68,4,true); //We ask for just
4 registers
        Gyr_rawX=Wire.read()<<8|Wire.read(); //Once
again we shif and sum
        Gyr_rawY=Wire.read()<<8|Wire.read();
        /*---X---*/
        Gyro_raw_error_x = Gyro_raw_error_x +
(Gyr_rawX/32.8);
        /*---Y---*/
        Gyro_raw_error_y = Gyro_raw_error_y +
(Gyr_rawY/32.8);
```

```
    if(i==199)
    {
        Gyro_raw_error_x = Gyro_raw_error_x/200;
        Gyro_raw_error_y = Gyro_raw_error_y/200;
        gyro_error=1;
    }
}
} //end of gyro error calculation
} //end of setup void
void loop ()
{
    instruksi();
    radio.write(& kirimz, sizeof(kirimz)); //Sending the
message to receiver
}
```

B. 2 Program Instruksi

```
void instruksi ()
{
    if (Total_angle_x >27)
    {
        Serial.println("KANAN");
        kirimz = 1;
    }
    else if (Total_angle_x<-27)
    {
        Serial.println("KIRI");
        kirimz = 2;
    }
    else if (Total_angle_y >27)
    {
        Serial.println("MAJU");
```

```
        kirimz = 3;
    }
    else if (Total_angle_y < -27)
    {
        Serial.println("MUNDUR");
        kirimz = 4;
    }
    else
    {
        Serial.println("tengah");
        kirimz = 0;
    }
}
```

