RANCANG BANGUN *PROTOTYPE* ROBOT PENGANTAR MAKANAN DAN OBAT DIMASA PANDEMI

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung

Disusun Oleh :

Gavin Allufi Yanno NIRM: 1051808

Muhammad Ihsan Zuhdi NIRM: 1051817

POLITEKNIK MANUFAKTUR NEGERI BANGKA BELITUNG

TAHUN 2022

LEMBAR PENGESAHAN

RANCANG BANGUN PROTOTYPE ROBOT PENGANTAR MAKANAN DAN OBAT DIMASA PANDEMI

Oleh:

Gavin Allufi Yanno/1051808

Muhammad Ihsan Zuhdi/1051817

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan Program Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing 1

Indra Dwisaputra, M.T.

Pembimbing 2

Aan Febriansyah, M.T.

Penguji 1

Muhammad Iqbal Nugraha, M.Eng.

Penguji 2

Yudhi, M.T.

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama Mahasiswa 1 : Gavin Allufi Yanno NIRM: 1051808
Nama Mahasiswa 2 : Muhammad Ihsan Zuhdi NIRM: 1051817

Dengan Judul : Rancang Bangun Prototype Robot Pengantar Makanan dan Obat Dimasa Pandemi

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 19 Januari 2022

Nama Mahasiswa Tanda Tangan

1. Gavin Allufi Yanno

2. Muhammad Ihsan Zuhdi

ABSTRAK

Sejak pandemi virus covid-19, banyak sekali tenaga medis yang menjadi korban akibat seringnya terjadi kontak langsung antara tenaga medis dengan pasien. Untuk mengurangi kontak tersebut, maka kami berinovasi membuat sebuah robot pembawa makanan dan obat yang bertugas untuk menggantikan tenaga medis dalam mengantar makanan dan obat ke pasien. Untuk pembacaan garis di lantai, sebuah sensor garis seri TCRT5000 digunakan. Robot pengantar makanan dan obat dalam penelitian ini berjalan dari pusat informasi ke kamar pasien dengan panduan garis di lantai, langkah pertama yang harus dilakukan adalah pengkalibrasian pada sensor, baik warna putih dan warna hitam, setelah itu barulah robot mengantar makanan dan obat ke kamar pasien. Sistem kontrol pergerakan mobilitas robot menggunakan sistem kontrol PID dengan penentuan nilai Kp, Ki dan Kd menggunakan metode trial and error. Hasil yang didapatkan dari proyek akhir ini adalah robot dapat mengantarkan makanan ke ruangan pasien dan berjalan dengan persentase keberhasilan tertinggi sebesar 90% menggunakan 8% dari kecepatan maksimal motor.

Kata Kunci: arduino, motor dc, pid, robot, sistem kontrol.

ABSTRACT

Since the covid-19 virus pandemic, many medical personnel have been victimized by frequent direct contact between medical personnel and patients. To reduce these contacts, we innovated to create a food and drug-carrying robot tasked with replacing medical personnel in delivering food and medicine to patients. The food and drug delivery robots in the study walked from the information center to the patient's room with a line guide on the floor with commands waving to the proximity sensor. Before the robot runs, the first step is calibration of the sensor, both puiih color and black, after which the robot delivers food and medicine to the patient's room. For line reading on the floor, a TCRT5000 series line sensor is used. The robot's mobility movement control system uses a PID control system with kp, ki and Kd value determination using trial and error methods. The result of this final project is that the robot can deliver food to the patient's room and walk with the highest percentage of success of 90% using 8% of the maximum speed of the motor.

Keywords: arduino, control system, dc motor, pid, robot.

KATA PENGANTAR

Assalamualaikum warahmatullahi wabarakatuh

Segala puji bagi Allah SWT yang telah memberikan kami kemudahan sehingga kami bisa menyelesaikan Laporan Proyek Akhir yang berjudul "Rancang Bangun Prototype Robot Pengantar Makanan dan Obat dimasa Pandemi". Shalawat serta salam juga tidak lupa kami curahkan untuk baginda Nabi Muhammad SAW yang akan memberikan syafaatnya di akhirat nanti.

Laporan Proyek Akhir ini dapat diselesaikan berkat adanya usaha dan kerja sama tim yang baik serta adanya bantuan dan dukungan dari berbagai pihak. Pada kesempatan ini penulis ingin mengucapkan terima kasih kepada pihak-pihak tersebut diantaranya:

1. Tuhan Yang Maha Esa atas segala rahmat, rezeki dan hidayah yang diberikan kepada penulis untuk menyelesaikan Proyek Akhir ini dengan baik dan tepat waktu.
2. Orang tua kami yang selalu memberikan dukungan dan doa yang sangat berharga.
3. Bapak I Made Andik Setiawan, M.Eng, Ph.D. Selaku Direktur di Politeknik Manufaktur Negeri Bangka Belitung yang telah memberikan kemudahan bagi kami untuk menyelesaikan Proyek Akhir ini.
4. Bapak Indra Dwisaputra, M.T selaku pembimbing 1 yang telah meluangkan waktu, tenaga dan pikiran untuk memberi pengarahan dalam penulisan Laporan Proyek Akhir.
5. Bapak Aan febriansyah, M.T selaku pembimbing 2 yang telah memberikan saran dan solusi selama proses perencanaan dan pembuatan alat serta penyusunan laporan.
6. Seluruh staf pengajar di Politeknik Manufaktur Negeri Bangka Belitung.
7. Rekan- rekan mahasiswa Politeknik Manufaktur Negeri Bangka Belitung.

8. Pihak-pihak lain yang telah memberikan bantuan selama proses pembuatan Proyek Akhir yang tidak bisa disebutkan satu persatu.

Dalam pembuatan Laporan Proyek Akhir ini, penulis menyadari bahwa Laporan ini masih jauh dari kata sempurna, baik dari segi isi maupun penulisannya. Oleh karena itu penulis mengharapkan kritik dan saran yang bersifat membangun demi kebaikan penulis kedepannya.

Akhir kata Penulis ucapkan terima kasih, semoga laporan ini dapat berguna dalam menambah pengetahuan dan wawasan untuk penulis dan pembaca.

Sungailiat, 2022

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1. Latar Belakang

Pandemi *Covid-19* masih menjadi permasalahan utama bagi masyarakat Indonesia. Sulitnya mencegah penyebaran kasus *Covid-19* di Indonesia terus menerus menjadi sorotan diberbagai media dan kalangan masyarakat. Dimasa pandemi seperti ini, tentu mematuhi dan menjalankan protokol kesehatan sangatlah penting. Salah satu protokol kesehatan yang selalu di himbau pemerintah adalah menjaga jarak.

Peran tenaga medis sangatlah penting dalam menangani pandemi saat ini. Jumlah tenaga medis yang minim mengharuskan mereka bekerja ekstra dan rela tidak bertemu dengan keluarga. Banyak sekali tenaga medis yang terpapar penyakit akibat seringnya kontak langsung dengan pasien.

Kami berinovasi untuk membuat robot pengantar makanan untuk pasien dimasa pandemi. Penggunaan robot ini diharapkan dapat meminimalisir kontak langsung antara tenaga medis dengan pasien sehingga kemungkinan tenaga medis yang terpapar penyakit berkurang. Ide pembuatan prototype robot ini bermula dari keprihatinan civitas akademika dikarenakan banyaknya tenaga medis yang terpapar virus covid-19.

Pada penelitian sebelumnya robot dibuat menggunakan kamera pixy2 CMU cam5 yang dimana sensor tersebut mendeteksi nomor pintu dengan warna merah, hijau dan biru, pada penelitian ini didapatkanlah hasil persentase keberhasilan di pagi hari 55%, siang hari 70% dan di malam hari 35%, penelitian ini dilakukan oleh Muhammad Hilmi (Muhammad Hanif Hilmy, 2020). Berdasarkan pembahasan sebelumnya maka di protek akhir kali ini dibuatlah robot pengantar makanan dan obat dengan menggunakan sensor garis, yang nantinya diharapkan persentasenya keberhasilannya besar.

1.2. Rumusan Masalah

1. Bagaimana cara mengatur kestabilan untuk gerak robot *line follower*?
2. Bagaimana menerapkan sistem kontrol PID pada sensor garis robot pengantar makanan dan obat ?

1.3 Batasan Masalah

1. Permukaan lantai rata atau tidak bergelombang.
2. Beban pada masing-masing motor sama, maksimal 3 kg untuk masing-masing motor.
3. Robot bergerak pada lintasan garis berwarna hitam dengan background cerah atau putih.

1.4. Tujuan

1. Mengantarkan makanan dan obat dengan tetap menjaga jarak antara tenaga medis dan pasien.
2. Mengatur kestabilan kecepatan motor DC dengan menggunakan PID.
3. Dapat menerapkan sistem kontrol PID untuk pengaturan kecepatan motor.

BAB II
DASAR TEORI

2.1. Penelitian Sebelumnya

Penelitian mengenai pembuatan robot pengantar makanan sudah pernah dilakukan seperti yang dilakukan oleh Muhammad Hilmi (Muhammad Hanif Hilmy, 2020), dimana di penelitian ini dia membuat robot pengantar obat di masa pandemi dengan menggunakan sensor kamera pixy CMUcam5 yang dapat mengirimkan informasi berupa data warna. Dimana hasil dari penelitian ini robot dapat melakukan pekerjaannya yaitu mengantarkan obat ke kamar pasien. Dengan persentase keberhasilan sebesar 55% dipagi hari, 70% disiang hari dan 35% di malam hari .

Penelitian selanjutnya dilakukan oleh Janis et al. (Janis *et al.*, 2014) yaitu pembuatan robot pengantar makanan dengan menggunakan sensor *proximity,* dimana keluaran dari sensor *proximity* pada penelitian ini berupa pulsa digital dengan logika *high* dan sensor *proximity* tersebut diletakkan di bagian bawah kerangka robot. Hasil dari penelitian ini robot berhasil mengantarkan makanan ke kamar pasien dengan waktu tempuh keseluruhan untuk ruang 1 37 detik dan ruang 2 58 detik.

2.2. Robot *Line Follower*

Robot *line follower* atau robot line *tracer* merupakan salah satu robot *mobile* yang sering dikembangkan dan sering digunakan dalam lomba atau penelitian seseorang. Sesuai dengan namanya robot *line follower* juga merupakan robot otonom yang mampu mengikuti jalur. Robot *line follower* biasanya menggunakan 2 macam sensor yaitu sensor LDR dan foto diode (Sigit *et al.*, 2019).

Robot *line follower* biasanya menggunakan motor dc sebagai penggerak untuk berpindah dari satu titik ke titik lainnya. Untuk sumbernya sendiri, robot *line follower* menggunakan baterai. salah satu komponen terpenting yang ada di robot *line follower* tentunya adalah sensor. Sensor yang nantinya bakal dikombinasikan dengan program untuk membuat robot line follower berjalan mengikuti garis (Sigit *et al.*, 2019).

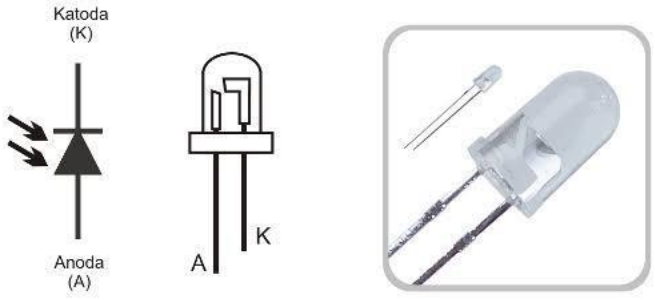
2.3. Sensor

Sensor merupakan alat yang berfungsi untuk mendeteksi atau mengukur sesuatu. Sensor biasanya digunakan untuk mengubah suatu mekanis, magnetis, sinar dan lain-lain menjadi arus atau tegangan. Prinsip kerja sensor adalah membuat perubahan nilai resistansi atau tahanan listrik, tegangan atau arus jika diberikan respon penghalang atau respon lainnya.

Variabel keluaran yang dihasilkan dari sensor berubah menjadi bentuk besaran listrik yang disebut tranduser. Jenis-jenis sensor sangatlah banyak dan dikelompokkan lagi menjadi beberapa, seperti sensor berdasarkan perubahan lingkungan ada 3 yaitu sensor kimia, sensor fisika dan sensor biologi. Setiap sensor pasti memiliki fungsi yang berbeda dan pemanfaatannya berdasarkan kegunaan atau kebutuhan.

2.3.1 Sensor Photodioda

Sensor photodiode merupakan jenis diode yang resistansinya akan berubah jika terkena sinar atau cahaya yang didapat dari transmitter atau *LED*. Resistansi dari photodioda bisa berubah-ubah tergantung dari banyaknya cahaya yang didapatkan. Semakin banyak cahaya yang diterima maka nilai resistansi akan semakin kecil, sebaliknya jika cahaya yang diterima kecil makan nilai resistansi akan semakin besar. Cara kerja sensor photodioda untuk *line follower* adalah *LED* akan mengirimkan cahaya ke garis lalu dipantulkan dan akan dibaca oleh photodioda (Adella, Kamal and Finawan, 2018).



Gambar 2.1 Simbol dan Bentuk Fisik Photodioda (Adella, Kamal and Finawan, 2018)

2.3.2 Sensor *Infrared Temperature* MLX90614

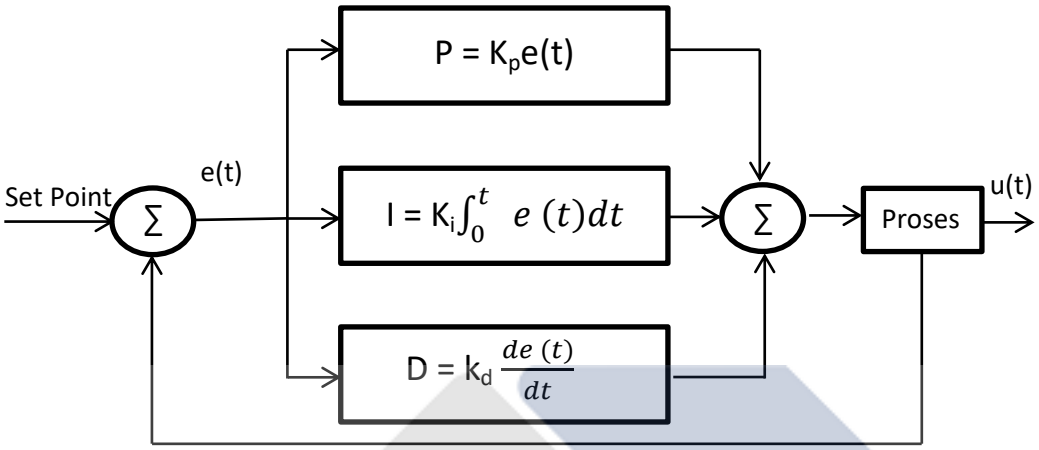
Sensor MLX90614 adalah sensor yang berguna untuk mengukur suhu dengan cara menggunakan radiasi gelombang inframerah. Sensor MLX90614 dapat secara otomatis mengkalibrasikan energi inframerah menjadi bentuk temperature. Sensor MLX90614 memiliki beberapa bagian yaitu *detector thermopile* inframerah MLX81101 dan *signal conditioning* ASSP MLX90303 yang berguna untuk memproses keluaran dari sensor. Sensor ini mampu mendeteksi hingga 70-80 celcius. Keluaran sensor berbentuk nilai digital karena sudah ada *ADC* di dalamnya. Prinsip kerja dari sensor MLX90614 adalah menangkap energy panas yang dihasilkan dari pancaran radiasi inframerah yang kemudian dikonversikan ke nilai besaran suhu (Mukhammad and Hyperastuty, 2021).

2.4. PID (Proportional Integral Derivatif)

Kontrol *Proportional Integral Derivatif* merupakan sebuah sistem kontrol yang membuat kontrol presisi dari suatu sistem dengan cara adanya umpan balik (*feedback*) pada sistem tersebut atau dengan kata lain sistem kontrol PID menggunakan nilai *error* yang didapatkan sebagai aksi untuk digunakan memperbaiki nilai *error* yang dibaca. Nilai *error* sendiri merupakan perbedaan dari *set point* yang telah ditentukan. Kontrol PID sangat berpengaruh untuk kestabilan kecepatan motor DC(Miftahul, Firdaus and Derisma, 2016). Sistem kontrol PID akan terus menerus menghitung nilai *error* dengan tujuan meminimalkan nilai *error* setiap waktu dengan penyetelan variabel kontrol (Jadmiko *et al.*, 2020).

$$u(t) = k_p\, e(t) + k_i \int_0^t e(t)\, dt + k_d \frac{de\,(t)}{dt} \quad \text{...} \quad (2.1)$$

Kp adalah kontrol *Proporsional* yang berfungsi memperbaiki respon transien khususnya nilai rise time dan seetling time. Ki adalah kontrol *Integral* yang punya sifat lebih lambat dari Kp. Kd adalah kontrol *Derivatif* yang dimana banyak akan berubah ketika ada perubahan *error*. Berikut adalah blok diagram kontrol PID (Joni, Ulum and Abidin, 2016).



Gambar 2. 2 Blok Diagram Kontroler PID (Joni, Ulum and Abidin, 2016)

Untuk mempermudah menentukan nilai Kp, Ki dan Kd bisa melihat tabel karakteristiknya, dengan melihat *output* dari sistem kontrol PID dapat dianalisa pengaruh nilai Kp, Ki dan Kd. Berikut tabel pengaruh nilai P, I dan D pada sistem kontrol (Anggraini, Ma'arif and Puriyanto, 2020).

Tabel 2.1 Pengaruh Nilai P, I dan D (Anggraini, Ma'arif and Puriyanto, 2020)

Nilai	Rise Time	Overshoot	Settling Time	Steady State *Error*
Proportional	Turun	Naik	Sedikit Perubahan	Turun
Integral	Turun	Naik	Naik	Menghilangkan
Derivatif	Sedikit Perubahan	Turun	Turun	Sedikit Perubahan

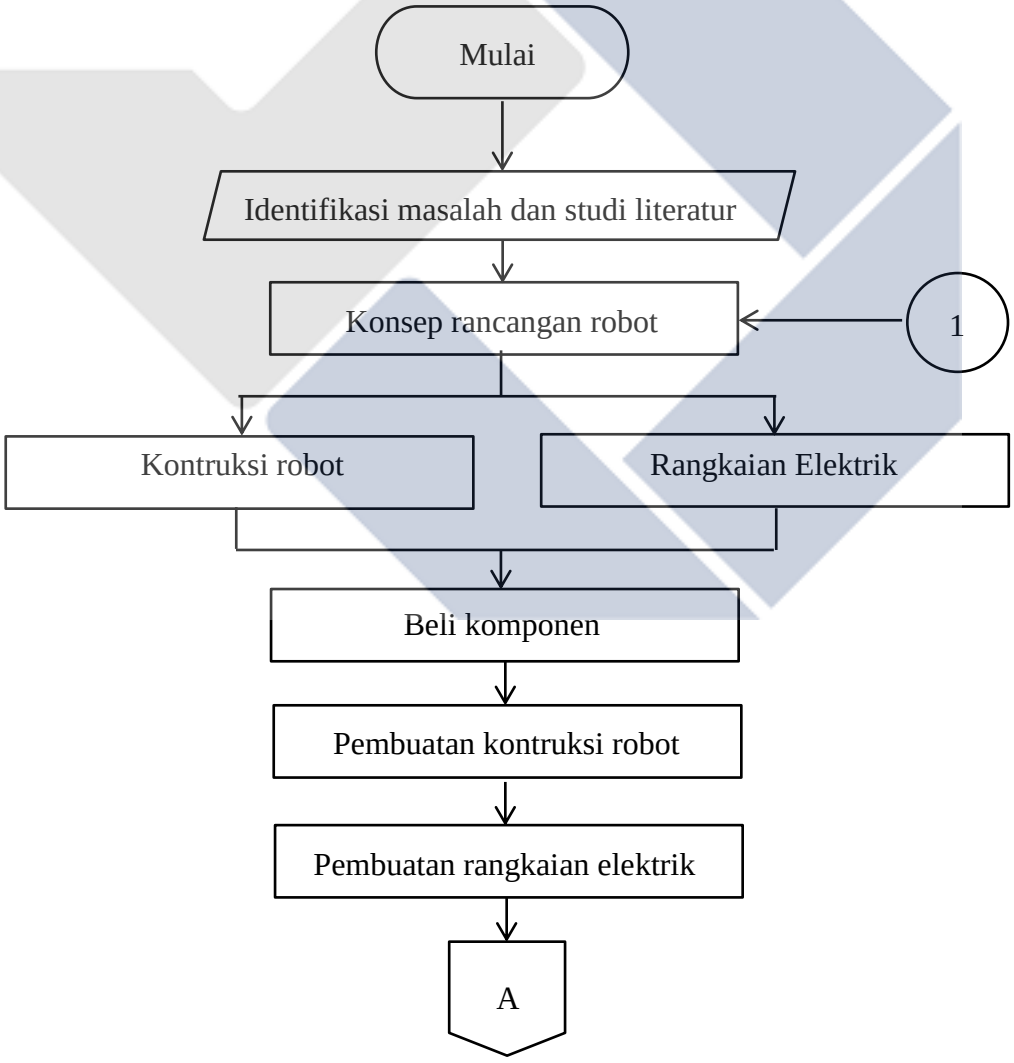
2.5. Platform MIT App Inventor

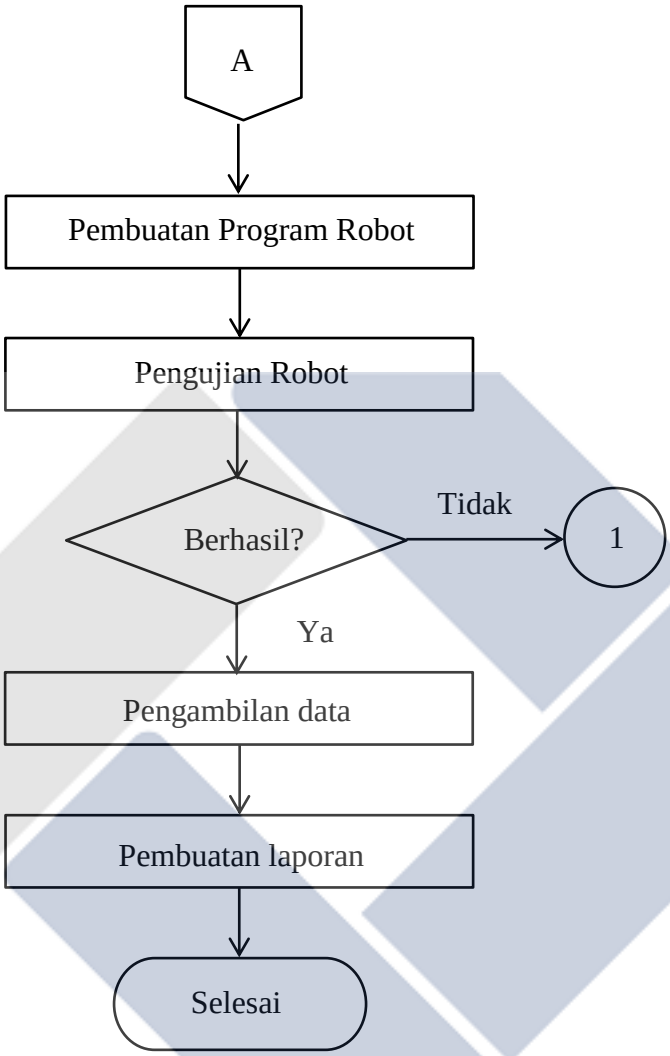
Google membuat sebuah program yang sangat baik yaitu App Inventor yang sekarang dikembangkan oleh MIT. Program ini biasa dimanfaatkan untuk membuat aplikasi di android yang terhubung dengan *web page* dan *java interface* (Widianto *et al.*, 2018). Untuk daftar dan menggunakannya hanya dengan menggunakan akun google yang ada di http://appinventor.mit.edu. Untuk membuat aplikasi pada *mobile phone* dibagi menjadi dua bagian yaitu :

1. App Inventor *Designer*, berguna untuk memilih konponen yang akan digunakan pada aplikasi.
2. App Inventor Blok Editor, berguna untuk merancang blok program yang nantinya blok diagram inilah menentukan bagaimana cara kerja dari aplikasi yang dibuat.

BAB III
METODE PELAKSANAAN

Selama pembuatan proyek akhir yang berjudul “Rancang Bangun Prototype Robot Pengantar Makanan dan Obat Dimasa Pandemi” terdapat beberapa tahapan proses yang bertujuan untuk mempermudah menyelesaikan proyek akhir ini. Metode pelaksanaan atau tahapan proses penngerjaan proyek akhir dapat dilihat pada gambar 3.1.





Gambar 3. 1 *Flowchart* Pembuatan Alat Proyek Akhir

Berdasarkan gambar diatas, proses pengerjaan proyek akhir ini dibuat melalui beberapa tahapan untuk penyelesaiannya. Berikut merupakan tahapan pembuatan proyek akhir ini

3.1. Studi Literatur

Pada tahap yang pertama, hal yang harus dilakukan adalah mencari dan mengumpulkan artikel maupun jurnal dari penelitian sebelumnya tentang robot pengantar makanan dan obat sebanyak mungkin. Studi literature adalah mempelajari

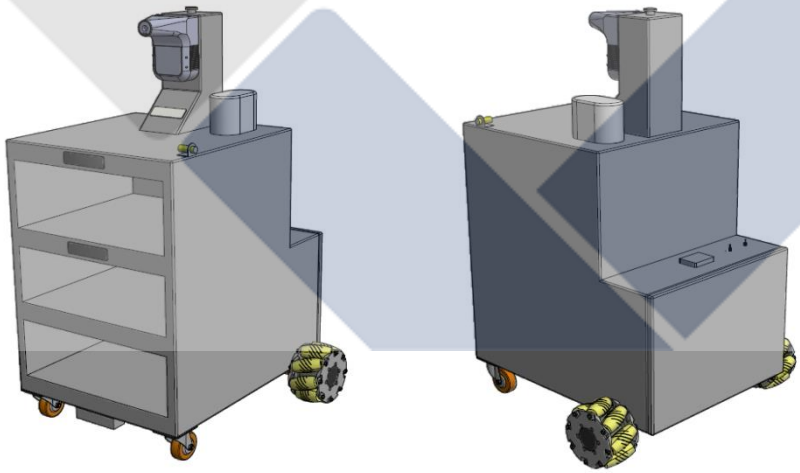
referensi-referensi yang bersumber dari buku maupun internet. Referensi-referensi yang dicari bertujuan agar dapat mengetahui prinsip kerja alat, komponen elektrik yang digunakan dan sistem kerja yang akan digunakan. Setelah beberapa data refensi dikumpulkan maka data tersebut yang akan dijadikan sebagai acuan dalam proses pembuatan proyek akhir.

3.2. Tahap Perancangan Alat

Tahap perancangan alat ini bertujuan untuk memberi gambaran mengenai alat yang akan dibuat. Pada tahap ini akan dibagi menjadi 2 yaitu perancangan *hardware* dan perancangan *software*.

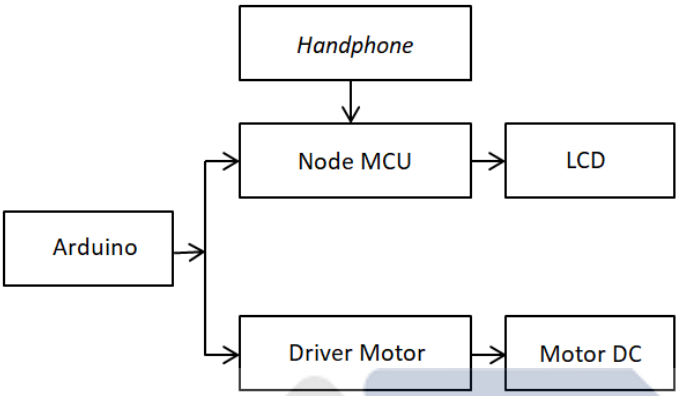
3.2.1. Perancangan Kontruksi

Perancangan kontruksi dimulai dari membuat desain kontruksi dan menentukan komponen-komponen yang akan digunakan untuk pembuatan alat proyek akhir seperti motor DC, sensor, Arduino dan lain-lain.



Gambar 3.2 Gambar Desain Kontruksi Alat

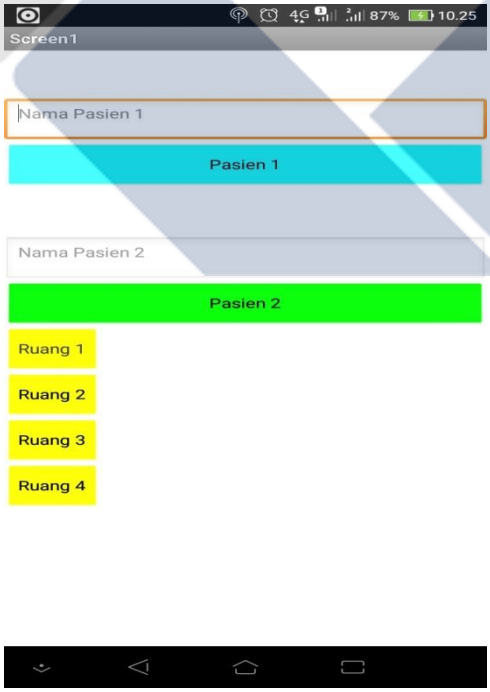
Setelah membuat desain kontruksi alat dan menentukan komponen, maka selanjutnya adalah membuat blok diagram robot. Blok diagram yang dibuat merupakan blok diagram dari kontrol robot, blok diagram inilah yang nantinya membantu memahami bagaimana kontrol robot bekerja.



Gambar 3. 3 Blok Diagram Kontrol Robot

3.2.2. Perancangan *Software*

Perancangan *software* adalah tahapan untuk merancang aplikasi yang nantinya memiliki beberapa fungsi yaitu, menjalankan dan memberhentikan robot, mengisi nama pasien pada lcd yang ada pada robot dan beberapa tombol untuk memilih ruangan yang akan diantarkan makanan dan obat. Berikut tampilan aplikasi layar monitor robot pengantar makanan dan obat:



Gambar 3. 4 Tampilan Layar Utama

3.3. Pembelian Komponen

Sebelum membuat kontruksi alat dan rangkaian eletriknya, tentu tahapan yang harus dilakukan adalah membeli komponen, alat dan bahan yang nantinya digunakan untuk membuat *prototype* robot pengantar makanan dan obat.

3.4. Pengujian Komponen Robot

Pengujian komponen-komponen robot bertujuan untuk mengetahui apakah komponen yang digunakan dapat bekerja dengan baik dan sesuai dengan fungsinya. Berikut uji coba komponen:

1. Uji coba Arduino mega.
2. Uji coba Sensor.
3. Uji coba Driver.
4. Uji coba Motor DC.

Setelah pengujian kompenen satu per satu, maka tahap selanjutnya adalah perakitan komponen secara keseluruhan sesuai dengan desain yang telah dibuat.

3.5. Pengujian *Software*

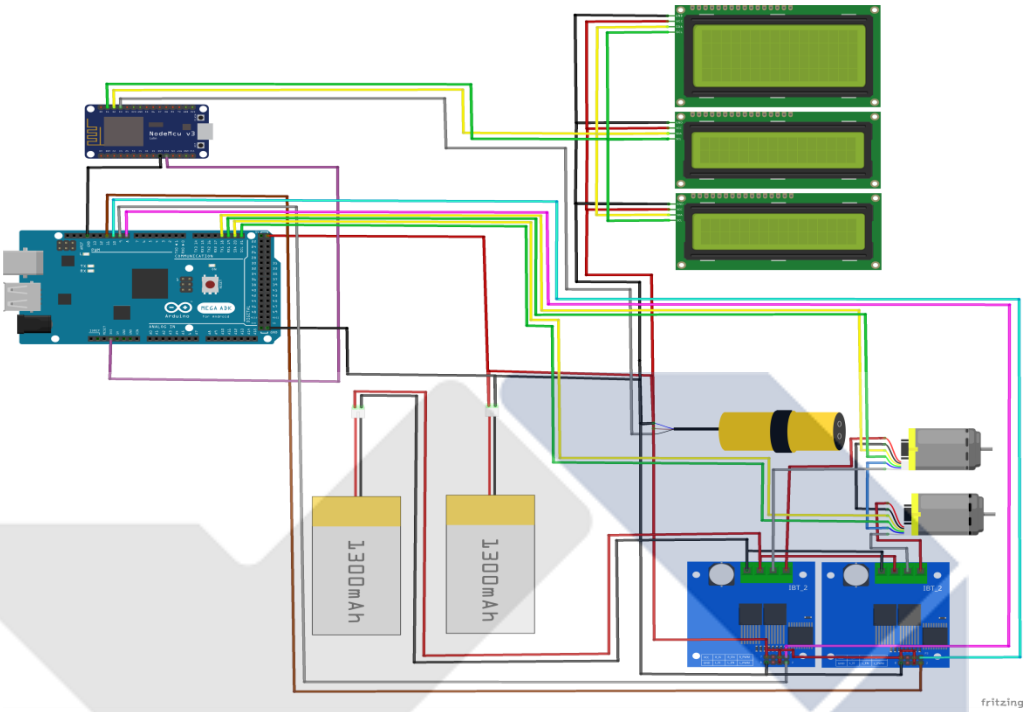
Pengujian *software* dilakukan apakah aplikasi yang telah dibuat menggunakan blynk bisa bekerja sesuai dengan yang diinginkan.

3.6. Pembuatan Kontruksi Alat

Pada tahapan ini dilakukan pembuatan kontruksi alat dengan acuan desain yang telah dibuat. Tahapan pembuatan kontruksi dimulai dari membuat base robot lalu dilanjutkan membuat kerangka atasnya menggunakan holo dan triplek yang sudah di cat.

3.7. Pembuatan Rangkaian Elektrik

Pada tahapan ini dilakukan pembuatan rangkaian elektrik sesuai dengan konsep yang telah dibuat. Tahapan pembuatan rangkaian elektrik yang dilakukan berupa pemasangan kabel jumper ke setiap motor yang nantinya dihubungkan ke driver dan arduino, serta pemasangan sensor garis yang nantinya juga dihubungkan ke arduino.



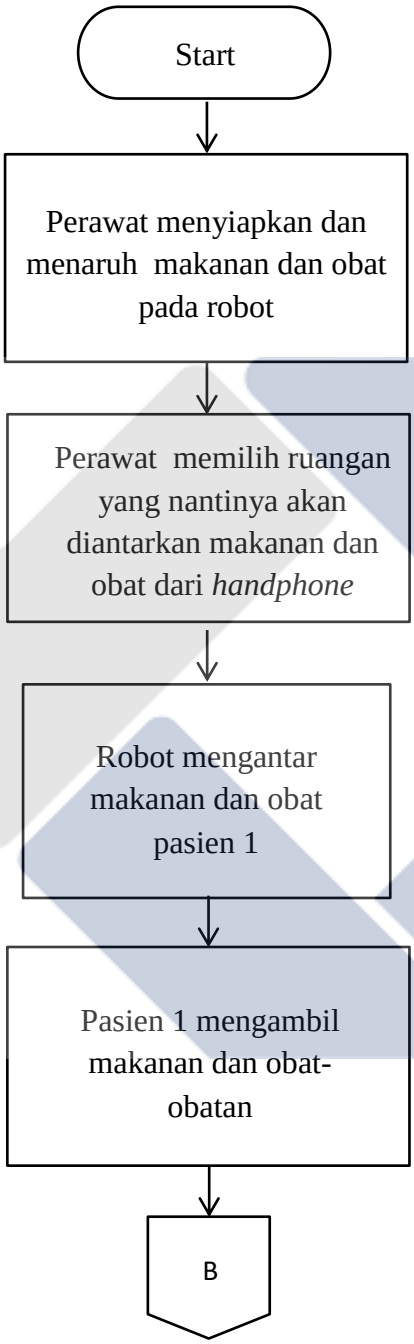
Gambar 3. 5 Skematik Rangkaian Elektrik

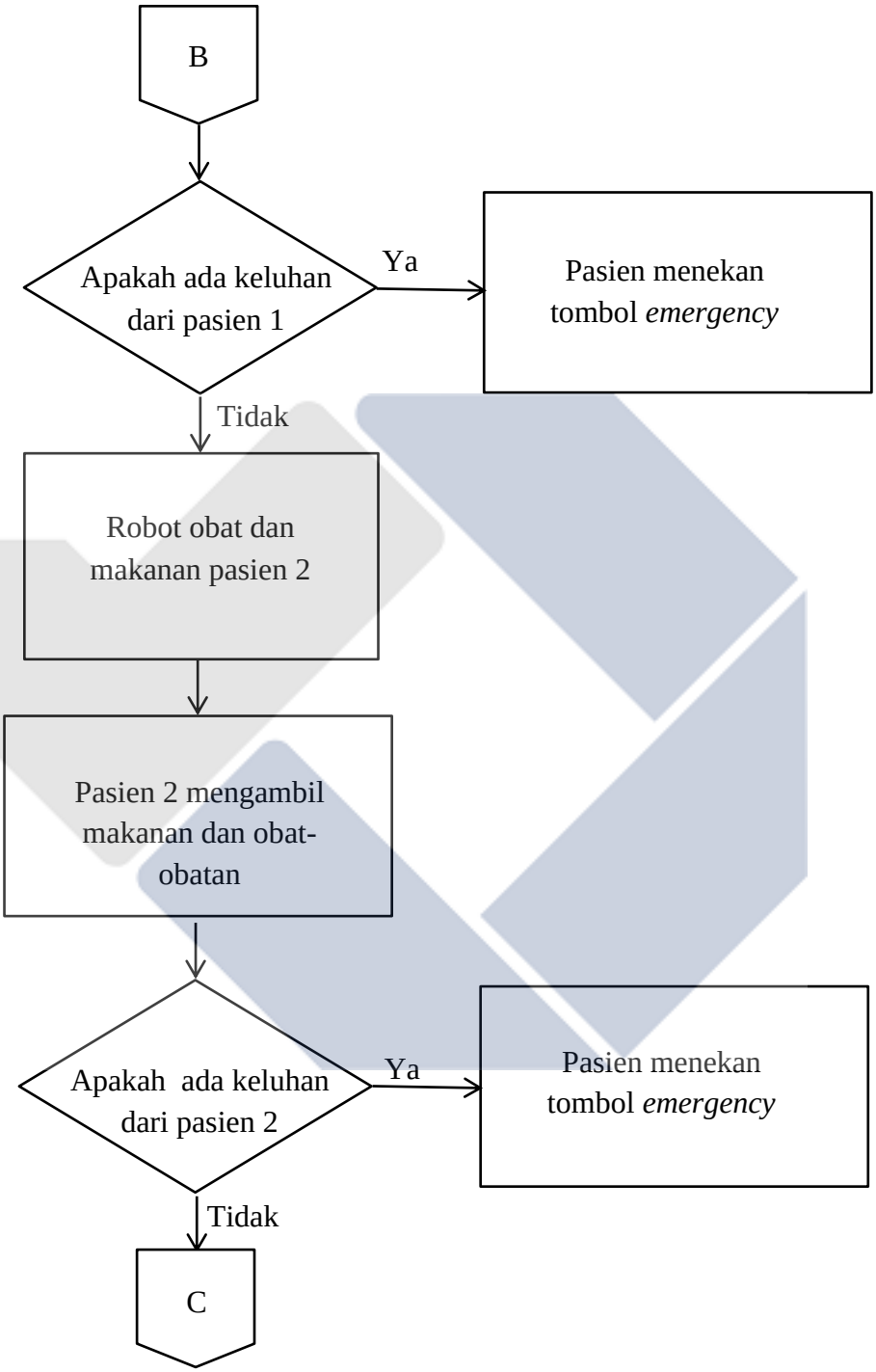
3.8. Pembuatan Program

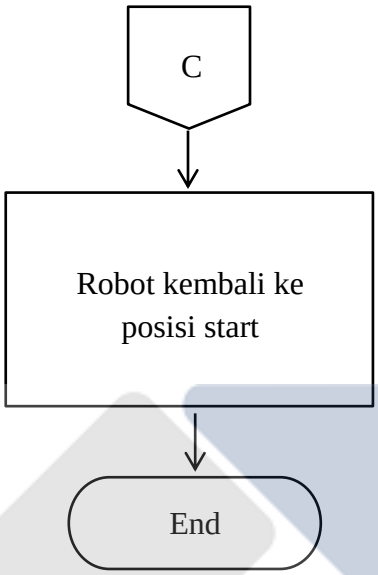
Pada tahapan ini dilakukan pembuatan program untuk sensor, PID, dan *Platform* MIT App Inventor yang menggunakan software Arduino IDE. Program-program tersebut akan dikombinasikan agar robot bekerja sesuai dengan konsep awal yang telah dirancang. Mulai dari robot yang mengantarkan makanan dan obat mengikuti garis sesuai dengan perintah yang diberikan operator dengan menggunakan sistem kontrol PID dan menampilkan nama-nama pasien pada LCD yang terpasang dirobot.

3.9. Pengujian Keseluruhan

Uji coba keseluruhan dilakukan untuk mengetahui hasil dari pengujian alat yang telah dibuat apakah bisa bekerja sesuai dengan yang diinginkan. Jika sudah berhasil maka nantinya data-data yang telah dikumpulkan akan dibuat kesimpulan dan saran untuk penelitian lebih lanjut.


Start
Perawat menyiapkan dan menaruh makanan dan obat pada robot
Perawat memilih ruangan yang nantinya akan diantarkan makanan dan obat dari *handphone*
Robot mengantar makanan dan obat pasien 1
Pasien 1 mengambil makanan dan obat-obatan
B


B
Apakah ada keluhan dari pasien 1
Ya
Pasien menekan tombol *emergency*
Tidak
Robot obat dan makanan pasien 2
Pasien 2 mengambil makanan dan obat-obatan
Apakah ada keluhan dari pasien 2
Ya
Pasien menekan tombol *emergency*
Tidak
C



Gambar 3. 6 Flowchart Pengujian Keseluruhan

3.10. Pengumpulan dan Analisa Data

Pada tahap ini merupakan tahapan dimana data yang dihasilkan alat akan dikumpulkan dan dianalisa yang bertujuan melihat kekurangan dan kelebihan alat yang telah dibuat dari segi kontrusi, sistem kontrol dan pengaplikasiannya ke lingkungan.

3.11. Pembuatan Laporan Proyek Akhir

Pada tahap ini merupakan tahapan akhir dalam pembuatan proyek akhir. Tahap ini bertujuan untuk merangkum keseluruhan dalam proses pembuatan proyek akhir dan memberikan informasi yang didapat dari alat yang telah dibuat.

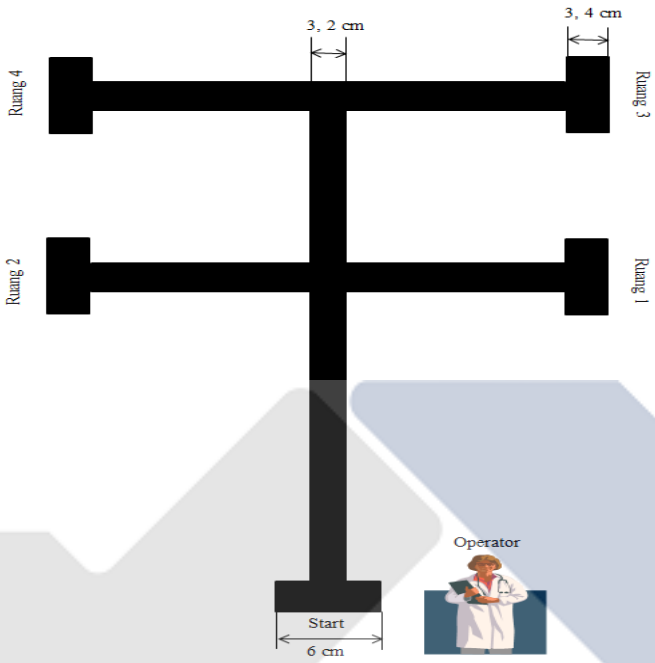
BAB IV
PEMBAHASAN

Pada bab ini akan membahas tentang tahapan dan metode yang digunakan dalam pembuatan serta pengujian robot pengantar makanan dan obat.

4.1. Deskripsi Alat

Robot pengantar makanan dan obat nantinya akan berjalan mengikuti garis yang telah dibuat. Robot akan berjalan stabil dikarenakan kontrol PID. Robot akan berhenti apabila robot telah berada di kamar pasien. Terdapat panduan untuk pasien sebelum mengambil makanan seperti cek suhu dan menggunakan *handsanitizer*, setelah itu barulah pasien mengambil makanan dan obat di laci. Jika pasien membutuhkan bantuan perawat maka pasien bisa menekan tombol emergency yang ada pada robot. Setelah pasien mengambil makanan pasien bisa melambaikan tangan di sensor *proximity.*

Mekanik robot pengantar makanan dan obat berupa holo dengan triplek yang dibentuk sesuai kerangka untuk menutupi holo tersebut. Untuk aktuator pada robot pembawa makanan dan obat menggunakan 2 motor DC dimana 2 roda karet dan 2 roda *free wheel.* Robot pembawa makanan dan obat menggunakan 8 sensor garis yang disusun pada bagian bawah robot tersebut. Di setiap laci akan ada nama pasien yang ditampilkan pada LCD. Untuk menginput nama pasien menggunakan aplikasi MIT app inventor yang ada pada handphone. Untuk sumber robot menggunakan 2 buah baterai lipo, satu baterai untuk motor DC dan baterai lainnya untuk kontrol, seperti arduino, lcd dan nodemcu. Bater untuk motor DC berkapasitas 22,2 V dan 4200mah sedangkan untuk kontrol menggunakan baterai yang berkapasitas 11,1 V dan 3600 mah, untuk sumber kontrol menggunakan *buck converter*. Untuk melakukan percobaan pada robot pengantar makanan dan obat, maka dibuatlah jalur untuk robot tersebut. Berikut gambar jalur untuk robot pengantar makanan dan obat.



Gambar 4. 1 Jalur Robot Pengantar Makanan dan Obat

Sebelum robot pengantar makanan dan obat dioperasikan, perawat harus menghidupkan robot terlebih dahulu, dimana saklar berada dibelakang robot. Selanjutnya perawat memilih ruangan mana yang akan diantarkan makanan dan obat melalui handphone dan perawat juga harus melakukan kalibrasi sensor garis secara berulang yang ada pada robot, kalibrasi dilakukan secara berulang dikarenakan untuk mengantisipasi *error* dari pembacaan sensor yang disebabkan oleh cahaya dan juga kalibrasi dilakukan secara bertahap, yang pertama kalibrasi warna putih dan yang kedua kalibrasi warna hitam, setelah kalibrasi dilakukan, barulah robot akan mengikuti jalur yang telah dibuat, bisa dilihat pada gambar diatas robot akan mulai pada posisi *start* dan kemudian akan menuju ruangan yang telah dipilih, perawat harus selalu siap apabila tombol *emergency* pada robot ditekan oleh pasien. Robot pengantar makanan dan obat memiliki tiga laci dimana laci pertama dan kedua (dihitung dari atas) untuk meletakkan makanan dan obat, dan laci ketiga untuk

meletakkan barang – barang. Setiap laci memiki lebar dan panjang sebesar 50 cm dengan tinggi 20 cm, dengan ukuran tersebut diharapkan dapat memudahkan perawat untuk meletakkan makanan, obat dan barang. Berikut gambar kontruksi robot pengantar makanan dan obat.

Gambar 4.2 Kontruksi Robot Pembawa Makanan dan Obat

4.2. Perencanaan dan Pembuatan Robot

Pada perencanaan pembuatan robot dibagi menjadi beberapa tahapan. Tahapan pertama adalah perencanaan *base* robot pengantar makanan dan obat. Tahapan yang kedua pemasangan roda. Tahapan yang ketiga adalah pembuatan kerangka robot pengantar makanan dan obat menggunakan. Tahapan keempat pemasangan komponen dan alat untuk melengkapi robot pengantar makanan dan obat seperti *handsanitizer*, *thermogun*, sensor dan lain-lain. Berikut tahapan yang dilakukan dalam proses pembuatan robot pembawa makanan dan obat :

a. Pembuatan Base Robot

Hal pertama yang harus dibuat saat membuat robot adalah membuat *base* untuk robot. Pembuatan *base* robot tentunya harus menggunakan perhitungan dan pertimbangan agar robot dapat bekerja sesuai dengan yang diinginkan. Adapun

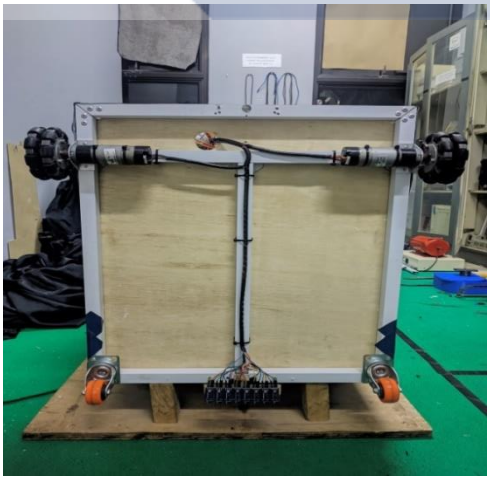
beberapa pertimbangan yang harus diperhatikan adalah beban robot itu sendiri, seberapa fleksibel robot bergerak dan kecepatan robot nantinya.

Setelah beberapa pertimbangan tersebut, langkah selanjutnya adalah menentukan komponen yang akan digunakan, setelah itu adalah proses *marking* lalu memotong holo dengan panjang sesuai desain robot. *Base* inilah yang nanti akan menjadi tempat untuk meletakkan beberapa komponen seperti motor DC dan roda . *Base* robot yang telah dibuat dapat dilihat pada Gambar 4.2.

Gambar 4.3 Base Robot

b. Pemasangan Roda

Pada tahap ini yaitu pemasangan roda pada *base* robot, roda yang dipasang sebanyak 4 buah, 2 buah roda *free wheels* didepan dan 2 buah roda karet dengan motor DC. Pemasangan roda sangat perlu diperhatikan agar ketika robot berjalan tidak mengalami kendala.



Gambar 4.4 Pemasangan Roda

c. Pembuatan Kerangka Robot

Pada tahap ini dibuatlah kerangka robot pembawa makanan dan obat. Kerangka robot memiliki peran sebagai penopang beban yang nantinya akan diletakkan pada robot. Kerangka robot dirakit menggunakan besi *hollow* dan dibentuk sesuai dengan gambar desain dan ditutupi menggunakan triplek.

Gambar 4.5 Kerangka Robot

Gambar 4. 6 Kerangka Robot Setelah Ditutupi Triplek

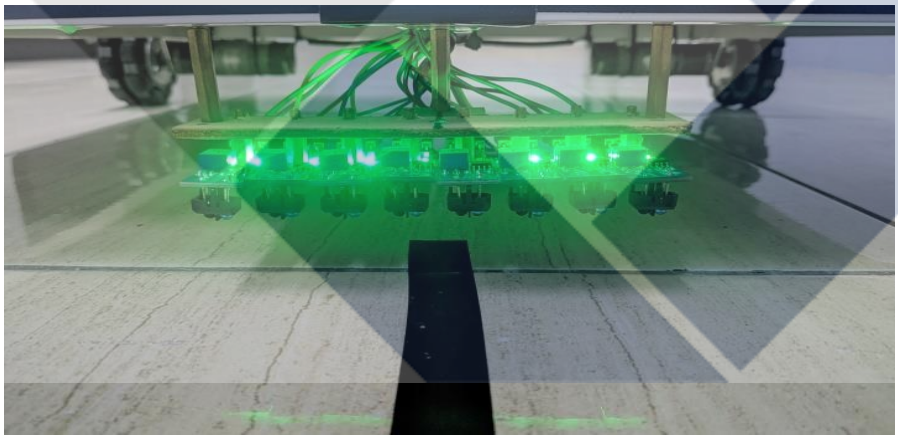
d. Pemasangan Komponen dan Alat Pada Robot

Pada tahap ini adalah memasang komponen dan alat di posisinya masing-masing, komponen yang dipasang pada robot tentunya memiliki fungsi masing-masing, pemasangan komponen dilakukan secara bertahap sesuai dengan kebutuhan

pada robot, mulai dari pemasangan sensor hingga pemasangan *handsanitizer*. Pada tahap ini juga dilakukan perakitan rangkaian elektrik atau *assembling* elektrik robot. *Assembling* elektrik robot dirakit sesuai dengan gambar rangkaian yang telah di desain, perakitan elektrik robot harus sangta diperhatikan agar tidak terjadi *short* pada rangkaian, karena apabila terjadi *short* pada rangkaian maka kemungkinan akan membuat komponen dan alat rusak. Perakitan elektrik robot juga harus bisa serapi mungkin karena jika terjadi *trouble* pada robot dapat memudahkan untuk menemukannya dan dapat mencari solusi secepat mungkin.

- Pemasangan Sensor Garis

Pemasangan sensor garis diposisikan dibawah robot berjumlah 8 buah, sensor yang digunakan adalah sensor TCRT5000. Sensor diposisikan pada depan robot, sensor TCRT5000 dijadikan sebagai *input* untuk gerak motor DC. Pemasangan sensor TCRT5000 bisa dilihat pada gambar 4.6.



Gambar 4.7 Pemasangan Sensor Garis.

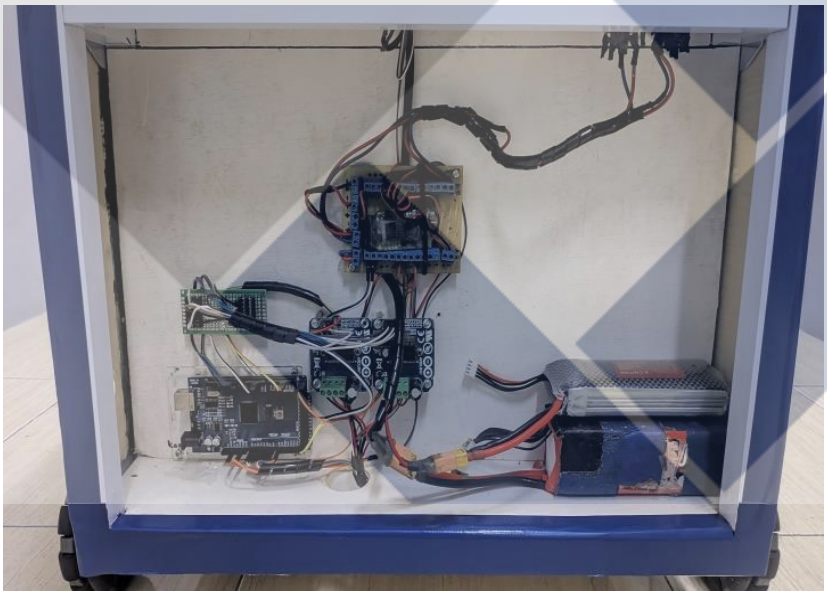
- Pemasangan Sensor *Proximity*

Pemasangan sensor diposisikan di atas robot, sensor *proximity* ini nantinya berfungsi sebagai *input.*

Gambar 4. 8 Pemasangan Sensor *Proximity*

- Pemasangan dan Perakitan Elektrik Komponen

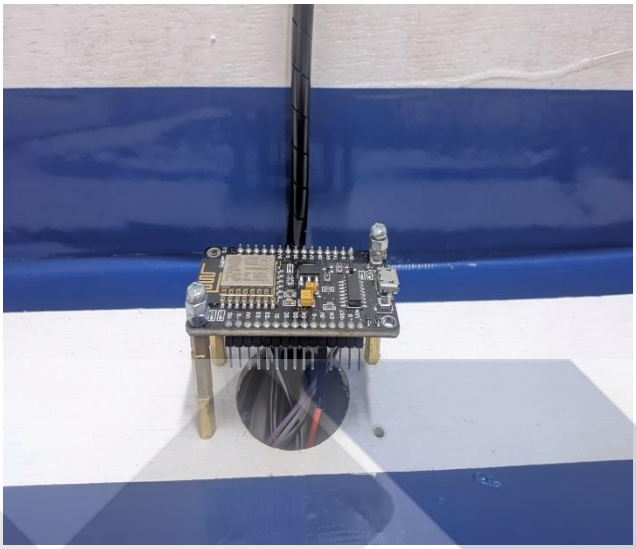
Pemasangan beberapa komponen ini diposisikan ditempat yang telah disediakan yaitu dibelakang robot. Komponen yang disusun disini adalah baterai, arduino, driver dan *buck converter*.



Gambar 4. 9 Pemasangan dan Perakitan Elektrik Robot

- Pemasangan NodeMCU

Pemasangan NodeMCU diletakkan dibelakang robot, NodeMCU ini nantinya akan menghubungkan *handphone* ke LCD. Dimana untuk input nama pasien bisa dilakukan dari *handphone* selama NodeMCU dan *handphone* terhubung dalam *wifi* yang sama.



Gambar 4.10 Pemasangan NodeMCU

- Pemasangan LCD

Pemasangan LCD diletakkan dibagian atas robot dan disetiap laci makanan dan obat untuk pasien. LCD yang paling atas untuk menginformasikan urutan yang harus dilakukan pasien sebelum mengambil makanan dan obat, sedangkan LCD yang ada di laci untuk memberikan informasi nama pasien.

Gambar 4.11 Pemasangan LCD Informasi Robot

Gambar 4.12 Pemasangan LCD Nama Pasien

- Pemasangan *Thermogun*

Pemasangan *thermogun* diletakkan dibagian atas robot agar dapat memudahkan pasien untuk menggunakannya.

Gambar 4. 13 Pemasangan *Thermogun*

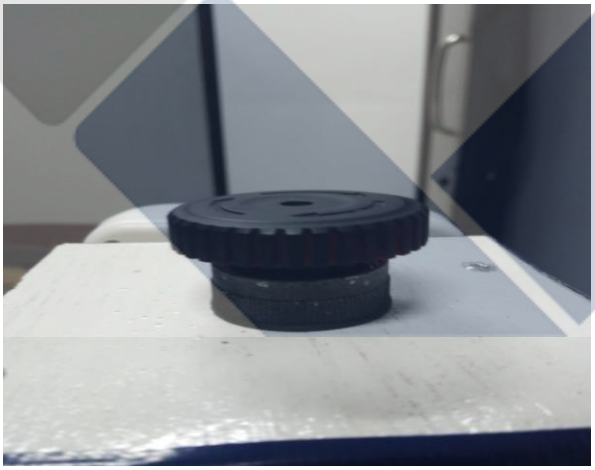
- Pemasangan *Handsanitizer*

Pemasangan *handsanitizer* diletakkan dibagian atas robot agar dapat memudahkan pasien untuk menggunakannya.

Gambar 4.14 Pemasangan *Handsanitizer*

- Pemasangan Tombol *Emergency*

Pemasangan tombol *emergency* diletakkan dibagian atas robot agar dapat memudahkan pasien untuk menggunakannya. Pasien bisa menekan tombol tersebut jika membutuhkan bantuan perawat.



Gambar 4.15 Pemasangan Tombol Emergency

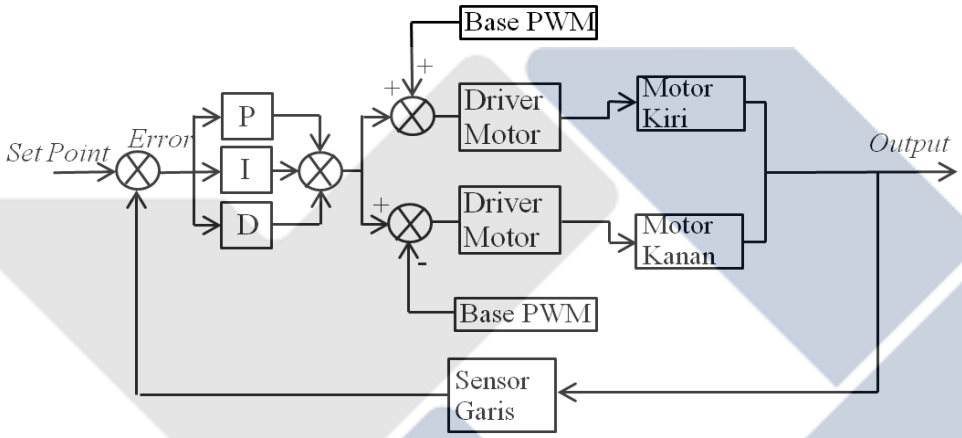
4.3. Pengujian Elektrik Robot

Pada tahap pengujian ini berguna untuk memastikan bahwa semua komponen dan alat robot dapat bekerja semestinya, pada tahap ini juga menggunakan beberapa

program pada arduino dan nodemcu untuk melakukan beberapa pengujian, seperti mencari persamaan linear antara PWM dan RPM dan pengujian sensor TCRT5000.

4.3.1. Blok Diagram

Blok diagram sangat diperlukan untuk mempermudah menentukan tahapan dari suatu sistem kontrol pada robot. Berikut blok diagram kontrol robot bisa dilihat pada gambar 4.16.



Gambar 4. 16 Blok Diagram Kontrol Robot

4.3.2. Pengujian Persamaan Linear antara PWM dan RPM

Sebelum pengujian persamaan linear antara PWM dan RPM dilakukan, tahapan yang harus dikerjakan terlebih dahulu adalah melihat perbedaan nilai RPM yang didapatkan dari *Tachometer* dengan nilai RPM hasil dari pembacaan sensor *encoder* yang ditampilkan pada serial monitor.

Tabel 4. 1 Nilai Perbandingan dari Tachometer dan sensor encoder

Encoder	*Tachometer*	Persentase *Error* (%)
0	0	0,00%
35,71	34,9	2,32%
75,89	75,4	0,65%

118,3	117,8	0,42%
160,71	160	0,44%
203,13	202,5	0,31%
243,3	243	0,12%
285,71	285,3	0,14%
328,13	326	0,65%
368,3	367,8	0,14%
410,71	407,4	0,81%
450,89	449,7	0,26%
493,3	492,4	0,18%
535,71	531,5	0,79%
573,66	570,8	0,50%
611,61	610,9	0,12%
645,09	640,9	0,65%
698,66	695,5	0,45%
Rata - rata *error*		0,50%

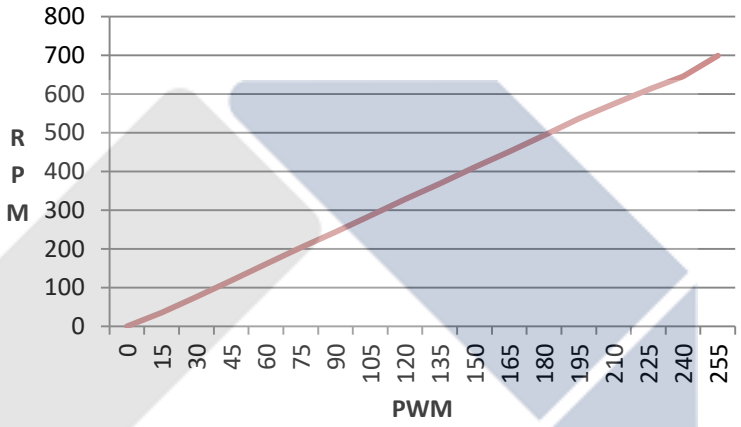
Dari data yang ada pada tabel 4.1 bisa dilihat bahwa rata – rata persentase *error* yang didapat dari nilai *tachometer* dan sensor *encoder* adalah 0,50%. Dari nilai tersebut bisa disimpulkan bahwa sensor *encoder* pada motor DC masih bisa digunakan.

Karena sensor *encoder* pada motor DC masih bisa digunakan maka tahap selanjutnya adalah mencari persamaan linear antar PWM dan RPM, nilai yang dihasilkan dari persamaan linear tersebut akan membantu dalam pemrograman sistem kontrol PID. Berikut Tabel dari hasil persamaan linear PWM dan RPM :

Tabel 4. 2 Hasil dari Persamaan linear PWM dan RPM

PWM(x)	RPM(y)	Grafik Persamaan Linear PWM dan RPM
0	0	

15	35,71
30	75,89
45	118,3
60	160,71
75	203,13
90	243,3
105	285,71
120	328,13
135	368,3
150	410,71
165	450,89
180	493,3
195	535,71
210	573,66
225	611,61
240	645,09
255	698,66



Dari tabel 4.2 diatas bisa didapatkan nilai persamaan linear PWM dan RPM, untuk rumus persamaan linear sebagai berikut :

$y = mx+c$ ..... (4.1)

$m = \Delta y/\Delta x = (y2-y1)/(x2-x1)$ ..... (4.2)

Dari rumus diatas, dapatlah nilai gradien (m) = 2,741 dan c = -2,965, setelah mendapatkan nilai gradien (m) dan nilai konstanta (c) maka rumus persamaan linear PWM dan RPM bisa ditulis sebagai berikut :

$RPM = m(PWM)+b$ atau $(RPM – b)/m = PWM$..... (4.3)

Persamaan diatas nantinya akan digunakan untuk menyinkronkan nilai PWM dengan RPM pada kontrol PID.

4.3.3. Pengujian Sensor Garis TCRT5000

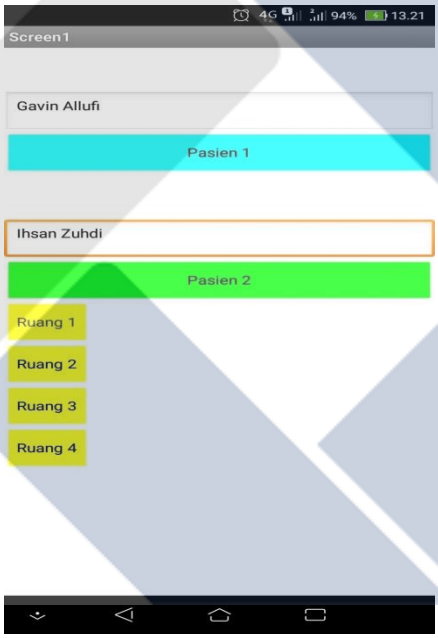
Pada pengujian sensor ini dilakukan untuk membuktikan bahwa sensor dapat bekerja sesuai dengan yang diinginkan, hal pertama yang harus dilakukan adalah kalibrasi sensor, setelah dikalibrasi kemudian adalah *mapping* pada sensor untuk mendapatkan nilai proses variabel. Berikut tabel hasil *mapping* nilai proses variabel :

Tabel 4. 3 Hasil Mapping Nilai Sensor

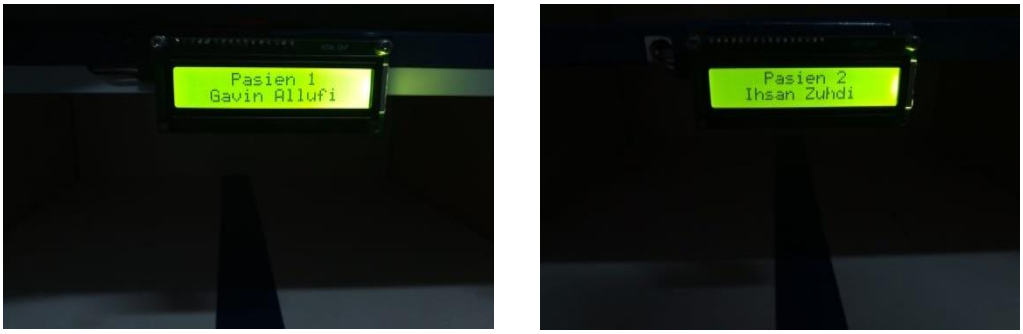
Case	Nilai Posisi
0b00000001	-10
0b00000111	-9
0b00000011	-8
0b00000010	-7
0b00001110	-6
0b00000110	-5
0b00000100	-4
0b00011100	-3
0b00001100	-2
0b00001000	-1
0b00011000	0
0b00010000	1
0b00110000	2
0b00111000	3
0b00100000	4
0b01100000	5
0b01110000	6
0b01000000	7
0b11000000	8
0b11100000	9
0b10000000	10

4.4. Pengujian Aplikasi MIT App Inventor

Setelah mengunduh aplikasi yang telah dibuat, maka aplikasi tersebut bisa langsung digunakan dan dicoba, pengujian ini bisa dilakukan dengan cara menghubungkan NodeMCU dan *Handphone* ke dalam satu wifi yang sama. Perubahan tampilan LCD terkadang masih mengalami *delay,* hal ini bisa dikarenakan sinyal wifi yang digunakan, faktor lainnya adalah pengiriman ke thinkspeak dan penerimaan data ke LCD. Berikut tampilan aplikasi MIT App Inventor dan LCD pada saat pengujian.



Gambar 4.17 Tampilan Aplikasi MIT App Inventor

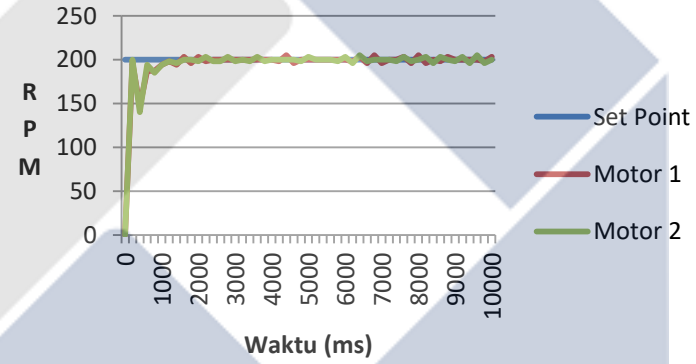
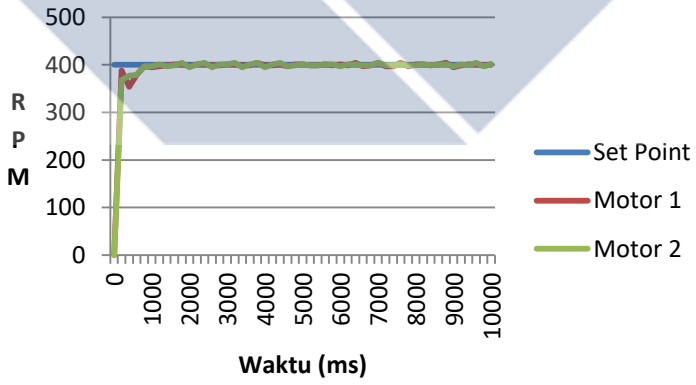


Gambar 4.18 Tampilan LCD Nama Pasien

4.5 Pengujian Sistem Kontrol PID Terhadap Motor DC

Pengujian sistem kontrol PID terhadap motor DC dilakukan untuk memastikan fungsi kontrol PID berguna bagi motor DC, dimana yang diketahui bahwa kontrol PID bisa mengurangi nilai *error*, sehingga kecepatan motor DC bisa lebih stabil. Di pengujian ini untuk menentukan nilai Kp, Ki dan Kd mengikuti tabel parameter yang ada di bab 2. Berikut tabel pengujian sistem kontrol PID untuk nilai RPM 200, 400 dan 600

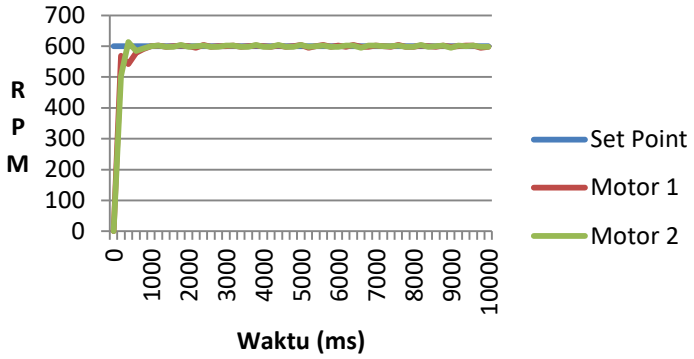
Tabel 4. 4 Perbandingan Hasil Pengujian Kontrol PID Terhadap Motor DC

RPM	Nilai	Hasil	Grafik
200	Kp = 0,5 Ki = 0,75 Kd =0,03	Overshoot = 0% Rise Time = 300ms Settling time = 2000ms Steady State = 5 RPM	
400	Kp = 0,3 Ki = 0,88 Kd = 0,002	Overshoot = 0% Rise Time = 800ms Settling time = 1400ms Steady State = 4 RPM	
600	Kp = 0,2 Ki = 0,88	Overshoot = 2,6% Rise Time = 700ms	

Kd = 0,02

Settling time = 1200ms

Steady State = 2 RPM



Melihat dari hasil pengujian kontrol PID yang ada pada tabel 4.3 bisa dikatakan pengujian yang ketiga dengan RPM 600 lebih baik dibandingkan dengan pengujian lainnya, hal ini dikarenakan nilai *steady state error* nya paling kecil yaitu 2 RPM, pengujian ini dilakukan dengan cara memberikan nilai Kp, Ki dan Kd dengan menggunakan metode *trial and error*. Dimulai dengan menentukan nilai Kp terlebih dahulu sampai dengan yang terakhir yaitu nilai Kd.

4.6 Pengujian Keseluruhan

Pengujian ini dilakukan untuk melihat apakah robot dapat bekerja sesuai dengan yang diharapkan. Berikut tabel pengujian robot pengantar makanan dan obat.

Tabel 4. 5 Data Pengujian Keseluruhan

Base PWM	Percobaan (kali)	Keberhasilan	Persentase (%)
20	10	9	90
25	10	7	70
30	10	7	70

Berdasarkan data diatas, dapat dilihat tingkat keberhasilan yang paling besar pada saat menggunakan base PWM 20, hal ini bisa dikarenakan pembacaan sensor yang dipengaruhi dari kenaikan nilai base PWM. Berikut tabel data waktu tempuh robot pengantar makanan dan obat.

Tabel 4. 6 Data Waktu Tempuh Robot Mengantarkan Makanan dan Obat

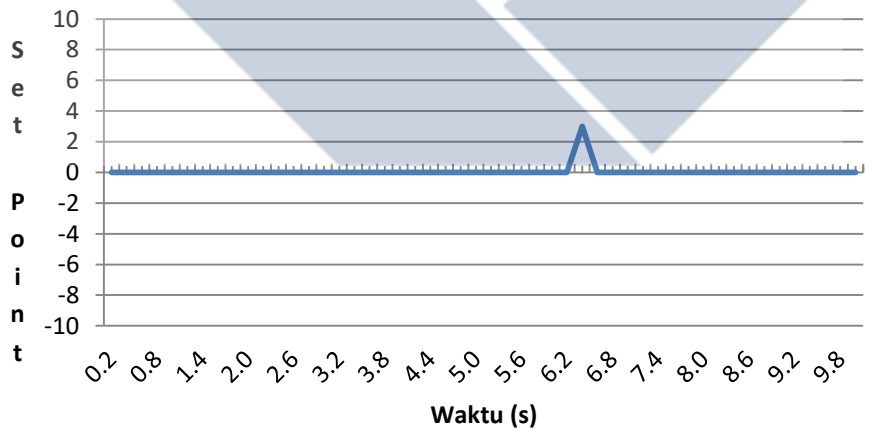
BasePWM	Waktu ke Ruang 1	Waktu ke Ruang 2
20	13 detik	27 detik

Dari tabel diatas bisa dilihat waktu robot dari posisi *start* ke ruangan 1 dan ke ruangan 2, waktu ke ruangan 2 yang ada pada tabel adalah waktu dari *start* ke ruang 1 lalu ke ruang 2 dengan waktu pasien saat mengambil makanan dan obat pada ruang 1 diabaikan.

4.7 Pengujian Kontrol PID Terhadap Sensor Garis

Pengujian ini dilakukan untuk melihat seberapa besar pengaruh nilai PID terhadap pembacaan sensor garis dengan *setpoint* yaitu 0, berikut tabel pengujian kontrol PID terhadap sensor garis pada robot pengantar makanan dan obat.

Tabel 4. 7 Pengujian Kontrol PID Terhadap Sensor Garis

Base PWM	Grafik Hasil Percobaan
20	

Data pengujian diatas menggunakan nilai KP = 20, KI = 0,1 dan KD = 20. Berdasarkan gambar diatas dapat dilihat bahwa robot berjalan cukup stabil dimana hanya terjadi sekali *error* pada detik 6,2 dan kembali stabil pada detik 6,6 detik.

BAB V
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan data dari hasil pengujian yang telah dilakukan, maka dapat disimpulkan sebagai berikut:

1. Penentuan nilai variabel Kp, Ki dan Kd menggunakan metode *trial and error* karena keterbatasan waktu dan hasil yang didapatkan sesuai dengan yang diinginkan.
2. Kontrol PID mampu memperbaiki nilai *error*, dimana nilai *error* yang didapatkan akan dikelola kembali dengan perhitungan PID itu sendiri sampai nilai *output* yang dihasilkan sudah mendekati atau sama dengan nilai *setpoint.* Dapat dilihat dari data yang ada dibab 4 dengan RPM 600 setelah menggunakan PID maka didapatkanlah nilai *steady state error* sebesar 2 RPM dan data pada pengujian kontrol PID terhadap sensor dengan menggunakan base PWM 20 robot dapat berjalan cukup stabil dimana hanya terjadi sekali error pada detik 6,2.
3. Robot dapat mengantarkan makanan dan obat ke setiap ruangan mengikuti garis yang telah dibuat secara otomatis, sehingga tenaga medis tidak perlu masuk ke ruangan. Keberhasilan robot dapat dipengaruhi oleh intensitas cahaya sekitar yang nantinya mempengaruhi pembacaan sensor garis. Waktu tempuh robot dengan menggunakan base PWM 20 adalah 13 detik keruangan pertama dan 27 detik ke ruangan kedua.
4. Persentase keberhasilan terbesar dari 10 kali percobaan robot mengantarkan makanan dan obat adalah 90% dengan menggunakan base PWM 20.

5.2. Saran

Dari hasil proyek akhir yang telah dilaksanakan, masih terdapat beberapa kekurangan dan masih bisa untuk dilakukan pengembangan. Berikut beberapa saran yang perlu penulis sampaikan :

1. Pembuatan robot pengantar makanan dan obat sebaiknya dilengkapi dengan sensor jarak yang nantinya berfungsi untuk pendeteksi halangan, baik itu orang maupun benda.
2. Menyempurnakan sistem kontrol dengan menggunakan metode tunning yang lebih baik lagi.

DAFTAR PUSTAKA

Adella, A., Kamal, M. and Finawan, A. (2018) 'Rancang Bangun Robot Mobile Line Follower Pemindah Minuman Kaleng Berbasis Arduino', *Jurnal Tektro,* 2(2).

Anggraini, S. F., Ma'arif, A. and Puriyanto, R. D. (2020) 'Pengendali PID pada Motor DC dan Tuning Menggunakan Metode Differential Evolution (DE)', *TELKA - Telekomunikasi Elektronika Komputasi dan Kontrol,* 6(2), pp. 147–159. doi: 10.15575/telka.v6n2.147-159.

Jadmiko, S. W. *et al.* (2020) 'Komparasi Kinerja Kendali PID dan Logika Fuzzy pada Simulator Plant Orde Dua', *JTERA (Jurnal Teknologi Rekayasa),* 5(2), p. 237. doi: 10.31544/jtera.v5.i2.2020.237-246.

Janis, D. A. N. *et al.* (2014) 'Rancang Bangun Robot Pengantar Makanan Line follower', *E-Journal Teknik Elektro Dan Komputer,* 3(1), pp. 1–10.

Joni, K., Ulum, M. and Abidin, Z. (2016) 'Robot Line Follower Berbasis Kendali Proportional-Integral-Derivative (PID) Untuk Lintasan Dengan Sudut Ekstrim', *Jurnal Infotel*, 8(2), pp. 138–142. doi: 10.20895/infotel.v8i2.129.

Miftahul, H., Firdaus, F. and Derisma, D. (2016) 'Pengontrolan Kecepatan Mobile Robot Line Follower Dengan Sistem Kendali PID', *TELKA - Telekomunikasi, Elektronika, Komputasi dan Kontrol,* 2(2), pp. 150–159. doi: 10.15575/telka.v2n2.150-159.

Muhammad Hanif Hilmy (2020) 'Prototipe Robot Pengantar Obat Otomatis Di Rumah Sakit Pada Masa Pandemi'.

Mukhammad, Y. and Hyperastuty, A. S. (2021) 'Sensitivitas Sensor MLX90614 Sebagai Alat Pengukur Suhu Tubuh Tubuh Non-Contact Pada Manusia', *Indonesian Journal of Professional Nursing,* 1(2), p. 51. doi: 10.30587/ijpn.v1i2.2339.

Sigit, H. T. *et al.* (2019) ‘Pelatihan Pembuatan Robot Line Follower untuk Meningkatkan Pengetahuan Robotika pada Siswa SMK Negeri I Kramatwatu Coresponding Author : fitrakbar06@gmail.com’, 1, pp. 230–240.

Widianto, Y. *et al.* (2018) ‘Kendali Mobile Robot dengan Suara Menggunakan Android Smart Phone’, *Seminar Nasional Sistem Informasi (SENASIF) Fakultas Teknologi Informasi Universitas Merdeka Malang*, 2(1), pp. 1027–1033. Available at: https://jurnalfti.unmer.ac.id/index.php/senasif/article/view/147/122.

LAMPIRAN 1

DAFTAR RIWAYAT HIDUP

DAFTAR RIWAYAT HIDUP

1. **Data Pribadi**

Nama Lengkap	: Gavin Aluffi Yanno
Tempat & Tanggal Lahir	: Bandung, 5 Mei 2000
Alamat Rumah	: Kp. Senang Hati Kab, Bangka Barat Prov.Kep. Bangka Beitung
No Hp	: 0895604471093
Email	: reusjaggerjack@gmail.com
Jenis Kelamin	: Laki-laki
Agama	: Islam

2. **Riwayat Pendidikan**

   SD Muhammadiyah Muntok (2006 - 2012)

   SMP Negeri 1 Muntok (2012 – 2015)

   SMK Negeri1 Muntok (2015 – 2018)

3. **Pendidikan Non Formal**

   -

Sungailiat, 26 Januari 2022

Gavin Aluffi Yanno

DAFTAR RIWAYAT HIDUP

1. **Data Pribadi**

Nama Lengkap	: Muhammad Ihsan Zuhdi
Tempat & Tanggal Lahir	: Pangkalpinang, 23 Oktober 2000
Alamat Rumah	: Jl. Beringin raya no. 128 Kec. Gerunggang Kel. Bukit Merapin Prov.Kep. Bangka Beitung
No Hp	: 082269064819
Email	: muhammadihsanzuhdi68@gmail.com
Jenis Kelamin	: Laki-laki
Agama	: Islam

2. **Riwayat Pendidikan**

   SD Negeri 60 Pangkalpinang (2006 - 2012)

   SMP Negeri 2 Pangkalpinang (2012 – 2015)

   SMA Negeri 1 Pangkalpinang (2015 – 2018)

3. **Pendidikan Non Formal**

   -

Sungailiat, 26 Januari 2022

Muhammad Ihsan Zuhdi

LAMPIRAN 2

PROGRAM

PROGRAM TESTING *LINE FOLLOWER*

```
#include <Wire.h>
#include <EEPROM.h>
//-----------------------------------------------------------------------------------------
Define Pins Number
#define kendaliTombol A8
//-----------------------------------------------------------------------------------------
Indikator
#define indi1 8
#define indi2 9
#define indi3 10
//-----------------------------------------------------------------------------------------
Motor
/** Motor Kiri **/
#define DIR_KI1 7
#define DIR_KI2 6
/** Motor Kanan **/
#define DIR_KA1 5
#define DIR_KA2 4
/**** SENSOR ****/
const int pinSensor[8] = {A0, A1, A2, A3, A4, A5, A6, A7};
//-----------------------------------------------------------------------------------------
Global variable
unsigned int bacaAnalog[8],sensor[8], nilaiRef1[8], nilaiRef2[8],nilaiKalibrasi[8];
int i,z,a=0, x=0 ,y=0,simpang=0,kanan=0,kiri=0;
boolean biner[8];
/**** RUNNING LED ****/
int count = 0;
```

```
unsigned long previousMillis = 0;
double NilaiPosisi,PosisiSensor, outPID;
int Kec_Min = -20;
int Kec_Max = 20;

void setup()
{
 pinMode(kendaliTombol, INPUT);
 pinMode(indi1, OUTPUT);
 pinMode(indi2, OUTPUT);
 pinMode(indi3, OUTPUT);
 digitalWrite(kendaliTombol, HIGH);
 digitalWrite(indi1, HIGH);
 digitalWrite(indi2, HIGH);
 digitalWrite(indi3, HIGH);
 Serial.begin(9600);
 initial_motor();
 for (int i = 0; i < 8; i++)
 {
  pinMode(pinSensor[i], OUTPUT);
 }
 for (int i = 0; i < 8; i++)
 {
  digitalWrite(pinSensor[i], HIGH);
  delay(250);
 }
 Serial.println("Dekatkan tangan anda pada sensor untuk melakukan kalibrasi ");
 while (digitalRead(kendaliTombol)==HIGH)
  {
```

```
   }
  digitalWrite(indi1, LOW);
  digitalWrite(indi2, LOW);
  digitalWrite(indi3, LOW);
  a=1;
//--------------------------------------------------------------------------------Kalibrasi
//  kalibrasiPutih ();
  dataSensorPutih();
//  kalibrasiHitam ();
  dataSensorHitam();
  kalibrasi();
  for(i=0;i<8;i++)
  {
     nilaiKalibrasi[i]=EEPROM.read(i);
     Serial.print("|");
     Serial.print(nilaiKalibrasi[i]);
     Serial.print("|");
     if (i == 7) Serial.println(" ");
  }
}
void loop()
{
//  repeat:
//-------------------------------------------------------------------------------
KondisiAwal_Motomati
  Serial.println("GASSSSSS");
  initial_sensor();
  driveMotor(0, 0);
  RunningLED(100);
```

```
 if (digitalRead(kendaliTombol)==HIGH)
 {
 while (a==4)
  {
//-------------------------------------------------------------------------------Maju
      lineFollow();
      while(simpang==1 && y==0)
      {
       driveMotor(15,15);
       delay (500);
       y=1;
       simpang=0;
      }
//-------------------------------------------------------------------------------
simpangPertamaKanan
      while(simpang==1 && y==1)
      {
       driveMotor(0,0);
       delay(200);
       driveMotor(20,20);
       delay(1000);
       driveMotor(0,0);
       delay(200);
       while(y==1)
       {
//-------------------------------------------------------------------------------RobotBacaPosisi
         SensorPosition();
//-------------------------------------------------------------------------------Manupergass
         driveMotor(-20,20);
```

```
          if(PosisiSensor==0)
          {
            driveMotor(0,0);
            delay(200);
            y=2;
            simpang=0;
          }
         }
        }
//-------------------------------------------------------------------------PasienPertama
       while(simpang==1 && y==2)
       {
        driveMotor(0, 0);
        delay (200);
        driveMotor(15,15);
        delay (1000);
        driveMotor(0, 0);
        while (digitalRead(kendaliTombol)==HIGH && y==2)
          {
            driveMotor(0,0);
          }
        y=3;
        simpang=0;
       }
//-------------------------------------------------------------------------
Pasien1_SudahMengambilMakanan
       while(y==3)
       {
//-------------------------------------------------------------------------RobotBacaPosisi
```

```
        SensorPosition();
//-------------------------------------------------------------------------------Manupergass
        driveMotor(-15,15);
        if(PosisiSensor==0)
        {
          driveMotor(0,0);
          delay(500);
          y=4;
          simpang=0;
        }
       }
//-------------------------------------------------------------------------------XXX
       while(y==4)
       {
//-------------------------------------------------------------------------------RobotLurusSaja
        lineFollowV2();
//-------------------------------------------------------------------------------pasienKedua
        while(simpang==1 && y==4)
        {
          driveMotor(0,0);
          delay (200);
          driveMotor(15,15);
          delay (1000);
          driveMotor(0,0);
//-------------------------------------------------------------------------------
Pasien2_SudahMengambilMakanan
          while (digitalRead(kendaliTombol)==HIGH && y==4)
          {
            driveMotor(0, 0);
```

```
      }
      y=5;
      simpang=0;
     }
    }

    while(y==5)
    {
//-------------------------------------------------------------------------RobotBacaPosisi
     SensorPosition();
//-------------------------------------------------------------------------Manupergass
     driveMotor(15,-15);
     if(NilaiPosisi==0)
     {
      driveMotor(0,0);
      delay(500);
      y=6;
      simpang=0;
     }
    }
//-------------------------------------------------------------------------belokKanan
    while(kanan==1 && y==6)
    {
     driveMotor(15,15);
     delay (500);
     driveMotor(0,0);
     delay (200);
     while(y==6)
     {
```

```
          SensorPosition();
          driveMotor(-20,20);
          delay(300);
          if(PosisiSensor==0)
          {
           driveMotor(0,0);
           delay(200);
           kanan=0;
           y=7;
           simpang=0;
          }
        }
      }
//-------------------------------------------------------------------------------Finish
      while(simpang==1 && y==7)
      {
        driveMotor(0,0);
        delay(200);
        driveMotor(20,20);
        delay(500);
        driveMotor(0,0);
        delay(200);
        while(y==7)
        {
//-------------------------------------------------------------------------------RobotBacaPosisi
          SensorPosition();
//-------------------------------------------------------------------------------Manupergass
          driveMotor(20,-20);
          while(NilaiPosisi==0)
```

```
        {
          driveMotor(0,0);
        }
        y=8;
       }
      }
      while(y==8)
      {
       driveMotor(0,0);
      }
    }
  }
}
```

PROGRAM CONTROL PID

```
double Error = 0, SumError = 0, LastError = 0;
double BasePWM = 20; //130 - 160 (Kp 25 Kd 132) //140 -255 (Kp 43 Kd 145)
double Kp = 20;       // Proporsional // Methode Ziegler KP 40 = 25 /24 (100 - 160) 43
(<160)  //50 >30
float Ki = 0.1;    // Integral     // 1.9 //0.1 // 0.2   > 0.56
float Kd = 20;      // Diferensial  // Methode Ziegler = 132 (Kp 25) 145 (43) >400
double Ts = 1;       // Time sampling (waktu cuplik)

/**** Fungsi perhitungan PID ****/
void lineFollow()
{
 Serial.println(NilaiPosisi);
 robotPosition();
 int SetPoint = 0;                    // Setpoint yang diinginkan
 Error = SetPoint - NilaiPosisi;       // Error
```

```
  double Delta Error = Error - Last Error; // Delta Error (Selisih error sekarang e(t)
den    gan error sebelumya e(t-1))
  Sum Error += Last Error;              // Akumulasi error
  double P = Kp * Error;               // Kontrol proporsional
  double I = Ki * Sum Error * Ts;      // Kontrol integral
  double D = ((Kd / Ts) * Delta Error);  // Kontrol derivative
  Last Error = Error;                  // Error sebelumnya
  outPID = P + I + D;                  // Output PID
  double motorKi = BasePWM - outPID;    // Motor Kiri
  double motorKa = BasePWM + outPID;    // Motor Kanan
  if (motorKi > Kec_Max)motorKi = Kec_Max;
  if (motorKi < Kec_Min)motorKi = Kec_Min;
  if (motorKa > Kec_Max)motorKa = Kec_Max;
  if (motorKa < Kec_Min)motorKa = Kec_Min;
  driveMotor(motorKi, motorKa);
//  if (Error == 0)driveMotor(motorKi, motorKa);
  if (Error == 0)driveMotor(Kec_Max, Kec_Max);

}

void lineFollowV2()
{
  Serial.println(NilaiPosisi);
  robotPositionV2();
  int SetPoint = 0;                    // Setpoint yang diinginkan
  Error = SetPoint - NilaiPosisi;      // Error
  double Delta Error = Error - Last Error; // Delta Error (Selisih error sekarang e(t)
den    gan error sebelumya e(t-1))
  Sum Error += Last Error;             // Akumulasi error
```

```
 double P = Kp * Error;                // Kontrol proporsional
 double I = Ki * SumError * Ts;        // Kontrol integral
 double D = ((Kd / Ts) * DeltaError);  // Kontrol derivative
 LastError = Error;                    // Error sebelumnya
 outPID = P + I + D;                   // Output PID
 double motorKi = BasePWM - outPID;    // Motor Kiri
 double motorKa = BasePWM + outPID;    // Motor Kanan
 if (motorKi > Kec_Max)motorKi = Kec_Max;
 if (motorKi < Kec_Min)motorKi = Kec_Min;
 if (motorKa > Kec_Max)motorKa = Kec_Max;
 if (motorKa < Kec_Min)motorKa = Kec_Min;
 driveMotor(motorKi, motorKa);
//  if (Error == 0)driveMotor(motorKi, motorKa);
 if (Error == 0)driveMotor(Kec_Max, Kec_Max);
}
```

PROGRAM RUNNING LED AND MOTOR

```
void driveMotor(double KEC_KI, double KEC_KA) {

//-------------------------------------------------------------------------------/**    Motor Kiri **/
 if (KEC_KI < 0)
 {
  analogWrite(DIR_KI1, 0); analogWrite(DIR_KI2, abs(KEC_KI));
 }
 else if (KEC_KI > 0)
 {
  analogWrite(DIR_KI1, abs(KEC_KI)); analogWrite(DIR_KI2, 0);
 }
 else
```

```
  {
    analogWrite(DIR_KI1, 0); analogWrite(DIR_KI2, 0);
  }

//------------------------------------------------------------------------------------  /** Motor
Kanan **/
  if (KEC_KA < 0)
  {
    analogWrite(DIR_KA1, abs(KEC_KA)); analogWrite(DIR_KA2, 0);
     }
  else if (KEC_KA > 0)
  {
    analogWrite(DIR_KA1, 0); analogWrite(DIR_KA2, abs(KEC_KA));
  }
  else
  {
    analogWrite(DIR_KA1, 0); analogWrite(DIR_KA2, 0);
  }
}

void initial_motor() {
  pinMode(DIR_KI1, OUTPUT);
  pinMode(DIR_KI2, OUTPUT);
  pinMode(DIR_KA1, OUTPUT);
  pinMode(DIR_KA2, OUTPUT);
}

//-------------------------------------------------------------------------------------/****
RunningLED ****/
```

```
void RunningLED(int pause)
{
  for (int i = 0; i <= 7; i++) {
    pinMode(pinSensor[i], OUTPUT);
  }
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= pause) {
    previousMillis = currentMillis;
    if (count == 0) {
      digitalWrite(pinSensor[0], ~HIGH);
      for (int i = 1; i <= 7; i++) {
        digitalWrite(pinSensor[i], LOW);
      }
    }
    else if (count == 1) {
      digitalWrite(pinSensor[1], HIGH);
    }
    else if (count == 2) {
      digitalWrite(pinSensor[0], LOW);
      digitalWrite(pinSensor[2], HIGH);
    }
    else if (count == 3) {
      digitalWrite(pinSensor[1], LOW);
      digitalWrite(pinSensor[3], HIGH);
    }
    else if (count == 4) {
      digitalWrite(pinSensor[2], LOW);
      digitalWrite(pinSensor[4], HIGH);
    }
```

```
else if (count == 5) {
  digitalWrite(pinSensor[3], LOW);
  digitalWrite(pinSensor[5], HIGH);
}
else if (count == 6) {
  digitalWrite(pinSensor[4], LOW);
  digitalWrite(pinSensor[6], HIGH);
}
else if (count == 7) {
  digitalWrite(pinSensor[5], LOW);
  digitalWrite(pinSensor[7], HIGH);
}
else if (count == 8) {
  digitalWrite(pinSensor[6], LOW);
  digitalWrite(pinSensor[7], HIGH);
}
else if (count == 9) {
  digitalWrite(pinSensor[6], LOW);
}
else if (count == 10) {
  digitalWrite(pinSensor[6], HIGH);
}
else if (count == 11) {
  digitalWrite(pinSensor[7], LOW);
  digitalWrite(pinSensor[5], HIGH);
}
  else if (count == 12) {
  digitalWrite(pinSensor[6], LOW);
  digitalWrite(pinSensor[4], HIGH);
```

```
    }
    else if (count == 13) {
      digitalWrite(pinSensor[5], LOW);
      digitalWrite(pinSensor[3], HIGH);
    }
    else if (count == 14) {
      digitalWrite(pinSensor[4], LOW);
      digitalWrite(pinSensor[2], HIGH);
    }
    else if (count == 15) {
      digitalWrite(pinSensor[3], LOW);
      digitalWrite(pinSensor[1], HIGH);
    }
    else if (count == 16) {
      digitalWrite(pinSensor[2], LOW);
      digitalWrite(pinSensor[0], HIGH);
      count = -1;
    }
    count++;
    delay(5);
  }
}
```

PROGRAM SENSOR

```
void initial_sensor()
{
  for (int i = 0; i < 8; i++)
  {
    pinMode(pinSensor[i], INPUT);
```

```
  }
}
int dataKalibrasi()
{
  for(i=0;i< 8;i++)
  {
    nilaiKalibrasi[i]=EEPROM.read(i);
  }
}
int BacaSensor()
{
  initial_sensor();
  dataKalibrasi();
  int baca[8] = {0};
  int Sensor = 0;
  for (int i = 0; i < 8; i++)
  {
   if (analogRead(pinSensor[i])/4 >= nilaiKalibrasi[i])
   {
    Sensor = Sensor | (0b10000000 >> i);
   }
//   baca[i] = analogRead(pinSensor[i])/4;
//   if (baca[i] >= nilaiKalibrasi[i]) baca[i] = 1;
//   else baca[i] = 0;
//   Serial.print("|");
//   Serial.print(baca[i]);
//   Serial.print("|");
//   if (i == 7) Serial.println(" ");
  }
```

```
  return Sensor;
}

void BacaSensorAnalog()
{
  initial_sensor();
  for (int i = 0; i < 8; i++)
  {
   bacaAnalog[i] = analogRead(pinSensor[i])/4;
//   Serial.print("|");
//   Serial.print(bacaAnalog[i]);
//   Serial.print("|");
//   if (i == 7) Serial.println(" ");
  }
}

void robotPosition()
{
  int sensor = BacaSensor();
  switch (sensor)
  {
    case 0b11111111 : x=1; break;
    case 0b00011111 : kiri=1; break;
    case 0b00001111 : NilaiPosisi = -11; break;
    case 0b00000001 : NilaiPosisi = -10; break;
    case 0b00000111 : NilaiPosisi = -9; break;
    case 0b00000011 : NilaiPosisi = -8; break;
    case 0b00000010 : NilaiPosisi = -7; break;
    case 0b00001110 : NilaiPosisi = -6; break;
```

```
   case 0b00000110 : NilaiPosisi = -5; break;
   case 0b00000100 : NilaiPosisi = -4; break;
   case 0b00011100 : NilaiPosisi = -3; break;
   case 0b00001100 : NilaiPosisi = -2; break;
   case 0b00001000 : NilaiPosisi = -1; break;
   case 0b00011000 : NilaiPosisi = 0; break;
   case 0b00010000 : NilaiPosisi =  1; break;
   case 0b00110000 : NilaiPosisi =  2; break;
   case 0b00111000 : NilaiPosisi =  3; break;
   case 0b00100000 : NilaiPosisi =  4; break;
   case 0b01100000 : NilaiPosisi =  5; break;
   case 0b01110000 : NilaiPosisi =  6; break;
   case 0b01000000 : NilaiPosisi =  7; break;
   case 0b11000000 : NilaiPosisi =  8; break;
   case 0b11100000 : NilaiPosisi =  9; break;
   case 0b10000000 : NilaiPosisi =  10; break;
   case 0b11110000 : NilaiPosisi =  11; break;
   case 0b11111000 : kanan=1; break;
  }
}

void robotPositionV2()
{
  int sensor = BacaSensor();
  switch (sensor)
  {
   case 0b11111111 : x=1; break;
   case 0b00000100 : NilaiPosisi = -4; break;
   case 0b00011100 : NilaiPosisi = -3; break;
```

```
    case 0b00001100 : NilaiPosisi = -2; break;
    case 0b00001000 : NilaiPosisi = -1; break;
    case 0b00011000 : NilaiPosisi = 0; break;
    case 0b00010000 : NilaiPosisi =  1; break;
    case 0b00110000 : NilaiPosisi =  2; break;
    case 0b00111000 : NilaiPosisi =  3; break;
    case 0b00100000 : NilaiPosisi =  4; break;
  }
}


void SensorPosition()
{
  int sensor = BacaSensor();
  switch (sensor)
  {
    case 0b00011000 : PosisiSensor = 0; break;
    case 0b00000000 : PosisiSensor = 1; break;


  }
}
```

PROGRAM KALIBRASI

```
void kalibrasi ()
{
 while (digitalRead(kendaliTombol)==HIGH && a==3)
 {
   Serial.println("harap dekatkan tangan anda unntuk menyelsaikan alibrasi dan menjalankan robot ");
```

```
  for(int i=0;i<8;i++)
  {
    if(bacaAnalog[i] > nilaiRef2[i]) nilaiRef2[i] = bacaAnalog[i];
    if(bacaAnalog[i] < nilaiRef1[i]) nilaiRef1[i] = bacaAnalog[i];
  }
  delay(50);
  for(int i=0;i<8;i++)
  {
    nilaiKalibrasi[i]=(nilaiRef1[i]+nilaiRef2[i])/2;
    EEPROM.write(i,nilaiKalibrasi[i]);
  }
 }

 digitalWrite(indi3, HIGH);
 a=4;
 Serial.println("Kalibrasi DONE!!");
 delay(1000);
}
```

PROGRAM SENSOR HITAM

```
void dataSensorHitam()
{
  for(i=0;i< 8;i++)
  {
    nilaiRef2[i]=EEPROM.read(i);
    Serial.print("|");
    Serial.print(nilaiRef2[i]);
    Serial.print("|");
    if (i == 7) Serial.println(" ");
```

```
  }
  delay(1000);
}

void kalibrasiHitam ()
{
  Serial.println ("kalibrasi Hitam");
  delay(1000);
  while (digitalRead(kendaliTombol) == HIGH && a==2)
  {
    Serial.println("Berhasil masuk ke While Baca Hitam");
    BacaSensorAnalog();
    for(i=0;i<8;i++)
    {
      nilaiRef2[i]=bacaAnalog[i];
      EEPROM.write(i,nilaiRef2[i]);
      }
  }
  digitalWrite(indi2, HIGH);
  a=3;
  Serial.println("Kalibrasi Hitam DONE!!");
  delay(1000);
}
```

PROGRAM SENSOR PUTIH

```
void dataSensorPutih()
{
  for(i=0;i< 8;i++)
  {
```

```
    nilaiRef1[i]=EEPROM.read(i);
    Serial.print("|");
    Serial.print(nilaiRef1[i]);
    Serial.print("|");
    if (i == 7) Serial.println(" ");
  }
  delay(1000);
}

void kalibrasiPutih ()
{
  Serial.println ("kalibrasi Putih");
  delay(1000);
  while (digitalRead(kendaliTombol) == HIGH && a==1)
  {
    Serial.println("Berhasil masuk ke While baca Putih");
    BacaSensorAnalog();
    for(i=0;i<8;i++)
    {
      nilaiRef1[i]=bacaAnalog[i];
      EEPROM.write(i,nilaiRef1[i]);
    }
  }
  digitalWrite(indi1, HIGH);
  a=2;
  Serial.println("Kalibrasi putih DONE!!");
  delay(1000);
}
```

PROGRAM TESTING PID 2 MOTOR

```
#define rpwm1 8
#define lpwm1 9
#define rpwm2 10
#define lpwm2 11

const byte interruptPin1 = 2;
const byte dataPin1 = 3;
const byte interruptPin2 = 21;
const byte dataPin2 = 20;

//Encoder variables
unsigned long motorPulseCounter1 = 0;
unsigned long motorPulseCounter2 = 0;
unsigned int encoderMeasurementInterval = 200;

//Timing variable
unsigned long lastTime = 0;

//RPM variable
int rpm1 = 0;
int rpm2 = 0;

int data=0;

float motorSpeed1 = 0;
float motorSpeed2 = 0;

float kp1 = 0.5;
```

```
float ki1 = 0.7;
float kd1 = 0;

float kp2 = 0;
float ki2 = 0;
float kd2 = 0;

unsigned int sp = 50;

int error1 = 0;
int last_error1 = 0;
int sum_error1 = 0;

int error2 = 0;
int last_error2 = 0;
int sum_error2 = 0;

float pwmConvM1=0;
float pwmConvM2=0;

void setup() {
  Serial.begin(9600);
  pinMode(interruptPin1, INPUT_PULLUP);
  pinMode(interruptPin2, INPUT_PULLUP);
  pinMode(rpwm1, OUTPUT);
  pinMode(lpwm1, OUTPUT);
  pinMode(rpwm2, OUTPUT);
  pinMode(lpwm2, OUTPUT);
  attachInterrupt(digitalPinToInterrupt(interruptPin1),motorInterrupt1,RISING);
```

```
 attachInterrupt(digitalPinToInterrupt(interruptPin2),motorInterrupt2,RISING);
 millis();
}
void loop()
{
   while((data<=50))
   {
      if (millis()-lastTime > encoderMeasurementInterval)
      {
        lastTime += encoderMeasurementInterval;
        rpm1 = calculateRPM1(motorPulseCounter1);
        rpm2 = calculateRPM2(motorPulseCounter2);
        motorPulseCounter1 = 0;
        motorPulseCounter2 = 0;
        measure();

        error1 = sp - rpm1;
        error2 = sp - rpm2;
        sum_error1 = sum_error1 + error1;
        sum_error2 = sum_error2 + error2;
        motorSpeed1 = ((kp1*error1)+(ki1*sum_error1) + (kd1 * (error1-last_error1)));
        motorSpeed2 = ((kp2*error2)+(ki2*sum_error2) + (kd2 * (error2-last_error2)));
        pwmConvM1 = ((motorSpeed1 +2.97)/2.74);// rumus convert dari RPM to PWM
        pwmConvM2 = ((motorSpeed2 +2.97)/2.74);// rumus convert dari RPM to PWM
        analogWrite(rpwm1, pwmConvM1);
```

```
        analogWrite(lpwm1, 0);
        analogWrite(rpwm2, 0);
        analogWrite(lpwm2, pwmConvM2);
        if (pwmConvM1 > 255) pwmConvM1 = 255;
        else if (pwmConvM1 < 0) pwmConvM1 = 0;
        else if (pwmConvM2 > 255) pwmConvM2 = 255;
        else if (pwmConvM2 < 0) pwmConvM2 = 0;
        last_error1 = error1;
        last_error2 = error2;
        data++;
      }
   }
        analogWrite(rpwm1, 0);
        analogWrite(lpwm1, 0);
        analogWrite(rpwm2, 0);
        analogWrite(lpwm2, 0);

}
float calculateRPM1(int pulses1)
{
  float scale1 = 5.0;
  float val1 = ((float)pulses1 / 134.4) * scale1 * 60.0;
  return val1;
}
float calculateRPM2(int pulses2)
{
  float scale2 = 5.0;
  float val2 = ((float)pulses2 / 134.4) * scale2 * 60.0;
  return val2;
```

```
}
void measure()
{
  Serial.print(sp);
  Serial.print("\t");
  Serial.print(rpm1);
  Serial.print("\t");
  Serial.println(rpm2);
}

void motorInterrupt1()
{
  motorPulseCounter1 +=1;
}

void motorInterrupt2()
{
  motorPulseCounter2 +=1;
}
```

PROGRAM RPM MOTOR

```
#define rpwm1 10
#define lpwm1 11

const byte interruptPin = 20;
const byte dataPin = 21;

//Encoder variables
```

```
unsigned long motorPulseCounter = 0;
unsigned int encoderMeasurementInterval = 200;

//Timing variable
unsigned long lastTime = 0;

//RPM variable
float rpm = 0;

//PWM variable
unsigned int sp = 0;

void setup() {
  Serial.begin(9600);
  pinMode(interruptPin, INPUT_PULLUP);
  pinMode(rpwm1, OUTPUT);
  pinMode(lpwm1, OUTPUT);
  attachInterrupt(digitalPinToInterrupt(interruptPin),motorInterrupt,RISING);
  analogWrite(rpwm1, sp);
  analogWrite(lpwm1, 0);
  millis();
}
void loop()
{
    if (millis()-lastTime > encoderMeasurementInterval)
    {
      lastTime += encoderMeasurementInterval;
      rpm = calculateRPM(motorPulseCounter);
      motorPulseCounter = 0;
```

```
    measure();
   }
}
float calculateRPM(int pulses)
{
 float scale = 5.0;
 float val = ((float)pulses / 134.4) * scale * 60.0;
 return val;
}

void measure()
{
  Serial.println(rpm);
}

void motorInterrupt()
{
 motorPulseCounter +=1;
}
```

PROGRAM LCD NODEMCU

```
#include <ThingSpeak.h>                    // add librery
#include <LiquidCrystal_I2C.h>
#include <ESP8266WiFi.h>

LiquidCrystal_I2C lcd1(0x23, 16, 2);
LiquidCrystal_I2C lcd2(0x26, 16, 2);
LiquidCrystal_I2C lcd3(0x27, 20, 4);
```

```
WiFiClient  client;

unsigned long counterChannelNumber = 1627792;              // Channel ID
const char * myCounterReadAPIKey = "L66BLCJTWW66C3A0";        // Read API
Key
const char * myCounterWriteAPIKey = "UAJ4A0OSU57QXEHT";

const int FieldNumber1 = 1;
const int FieldNumber2 = 2;       // The field you wish to read
const int FieldNumber3 = 3;

String presentStr,previousStr,presentStr2,previousStr2 = " ";
String line1, line2,line3, line4;
const int digitalPin = D6;                          // ESP8266 Analog Pin ADC0 = A0
int sensorNewValue = 0;                              // value read from the pot
int sensorOldValue = 0;

void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(digitalPin , INPUT);
  Serial.begin(115200);
////////////////////////////--KetRobot--////////////////////////////////
  lcd3.begin();
  lcd3.backlight();
  lcd3.setCursor(2, 0);
  lcd3.print("ROBOT PENGANTAR");
  lcd3.setCursor(6, 1);
```

```
  lcd3.print("MAKANAN");
  lcd3.setCursor(9, 2);
  lcd3.print("DAN");
  lcd3.setCursor(8, 3);
  lcd3.print("OBAT");
//////////////////////////--Pasien1--////////////////////////////////
  lcd1.begin();
  lcd1.backlight();
  lcd1.setCursor(4, 0);
  lcd1.print("Pasien 1");
  lcd1.setCursor(0, 1);
  lcd1.print("");
//////////////////////////--pasien_2--////////////////////////////////
 lcd2.begin();
 lcd2.backlight();
 lcd2.setCursor(4, 0);
 lcd2.print("Pasien 2");
 lcd2.setCursor(0, 1);
 lcd2.print("");
//////////////////////////--delay_penampil--////////////////////////////////
 delay(500);
//////////////////////////--Conf_WiFi--////////////////////////////////
 Serial.println();
 WiFi.begin("gais", "aaaaaaaa");                // write wifi name & password
Serial.print("Connecting");
 while (WiFi.status() != WL_CONNECTED)
  {
    delay(100);
    Serial.print(".");
```

```
  }
  Serial.println();
  Serial.print("Connected, IP address: ");
  Serial.println(WiFi.localIP());
  ThingSpeak.begin(client);

}

//////////////////////////--LCD_1--////////////////////////////////
void Pasien1()
{
 presentStr = ThingSpeak.readStringField(counterChannelNumber, FieldNumber1,
myCounterReadAPIKey);
 if(presentStr != previousStr)
 {
    int z=presentStr.length();
    line1 = presentStr.substring(0, 16);
    int a=(16-z)/2;
    lcd1.clear();
    lcd1.setCursor(4, 0); // Move the cursor characters to the right and
    lcd1.print("Pasien 1");  // Print HELLO to the screen, starting at 0,0.
    lcd1.setCursor(a, 1);
    lcd1.print(presentStr);
    Serial.println(presentStr);
    previousStr = presentStr;
 }
}

//////////////////////////--LCD_2--////////////////////////////////
```

```
void Pasien2 ()
 {
 presentStr2 = ThingSpeak.readStringField(counterChannelNumber, FieldNumber2,
myCounterReadAPIKey);
 if(presentStr2 != previousStr2)
 {
   int z=presentStr2.length();
   line3 = presentStr2.substring(0, 16);
   int a=(16-z)/2;
   lcd2.clear();
   lcd2.setCursor(4, 0); // Move the cursor characters to the right and
   lcd2.print("Pasien 2");  // Print HELLO to the screen, starting at 0,0.
   lcd2.setCursor(a, 1);
   lcd2.print(presentStr2);
   Serial.println(presentStr2);
   previousStr2 = presentStr2;
 }
}

void loop()
{
Pasien1();
Pasien2();
//int A = ThingSpeak.readLongField(counterChannelNumber, FieldNumber3,
myCounterReadAPIKey);
//  Serial.println(A);
//  digitalWrite(LED_BUILTIN, A);
//  sensorOldValue = sensorNewValue;
//  sensorNewValue = digitalRead(digitalPin);          // read the analog in value
```

```
//  if(sensorOldValue != sensorNewValue)
//  {
//          ThingSpeak.writeField(counterChannelNumber , 4 ,sensorNewValue , myCounterWriteAPIKey);
//    Serial.println(sensorNewValue);
//  }
//  delay(50);
}
```