REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN
CIPTAAN

Dalam rangka pelindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan	:	EC00202117903, 30 Maret 2021
Pencipta		
Nama	:	**Wahyudi, Dr. Hendra Jaya, S.Pd., M.T. dkk**
Alamat	:	Jln. A.P. Pettarani, Makassar, SULAWESI SELATAN, 90222
Kewarganegaraan	:	Indonesia
Pemegang Hak Cipta		
Nama	:	**Universitas Negeri Makassar**
Alamat	:	Jln. A.P. Pettarani, Makassar, SULAWESI SELATAN, 90222
Kewarganegaraan	:	Indonesia
Jenis Ciptaan	:	**Modul**
Judul Ciptaan	:	**MODUL ROBOTIKA BERBASIS MIKROKONTROLLER DAN INTERFACE**
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	:	1 Januari 2021, di Makassar
Jangka waktu pelindungan	:	Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor pencatatan	:	000245010

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL

Dr. Freddy Harris, S.H., LL.M., ACCS.
NIP. 196611181994031001

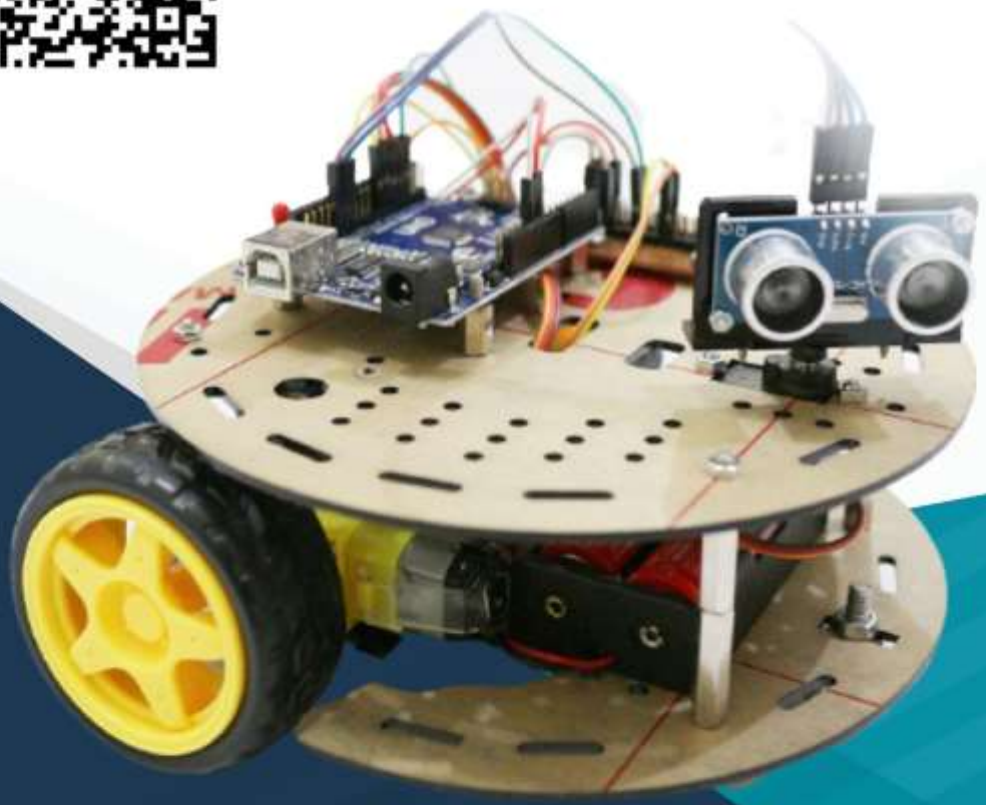
LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Wahyudi	Jln. A.P. Pettarani
2	Dr. Hendra Jaya, S.Pd., M.T.	Jln. A.P. Pettarani
3	Dr. Edy Sabara, M.Si.	Jln. A.P. Pettarani
4	Prof. Dr. Ir. H. Muhammad Yahya, M.Kes., M.Eng., IPU	Jln. A.P. Pettarani
5	Dr. Abdul Muis Mappalotteng, S.Pd., M.Pd., M.T.	Jln. A.P. Pettarani
6	Dr. Muh. Ma'ruf Idris, S.T., M.T.	Jln. A.P. Pettarani

MODUL ROBOTIKA

Trainer Robotika Berbasis Mikrokontroller Arduino UNO - Arduino Nano - ESP 8266

Hendra Jaya
Edy Sabara
Wahyudi



ARDUINO android IoT

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa yang telah memberi hidayah, kekuatan, kesehatan, dan ketabahan kepada kami sehingga penyusunan Modul Robotika berbasis Mikrokontroller dan Interface ini dapat terselesaikan.

Modul praktikum ini disusun dengan tujuan menyediakan materi dan panduan bagi peserta didik untuk menyelesaikan praktikum pada mata kuliah robotika. Modul praktikum ini diharapkan dapat mampu membantu peserta didik untuk belajar mandiri dan membuka kesempatan bagi setiap mahasiswa untuk belajar sesuai kecepatan masing-masing. Selain itu Modul praktikum ini diharapkan dapat mengembangkan kecakapan hidup dalam arti luas dan peningkatan wawasan peserta didik.

Materi dalam Modul praktikum ini diorganisasikan sesuai tingkat kesulitan masing-masing materi sehingga peserta didik mampu mempelajari setiap materi dengan lebih mudah. Kegiatan-kegiatan praktikum dikembangkan untuk menjadikan mahasiswa secara aktif belajar melalui kegiatan pemasangan rangkaian dan penyusunan program. Dalam pembelajaran, dosen diharapkan bertindak sebagai fasilitator, pemberi umpan balik, dan pendorong mahasiswa agar berani menggunakan teknik yang berbeda-beda untuk memecahkan masalah tertentu berdasarkan latar belakang pengetahuan dan kebiasaan masing-masing.

Penyusunan Modul praktikum ini dapat terselesaikan atas dukungan dari berbagai pihak. Oleh karena itu, kami mengucapkan terima kasih kepada semua pihak yang telah membantu dalam menyelesaikan Modul praktikum ini.

Walaupun kami telah menyusun Modul praktikum ini dengan upaya yang sungguh-sungguh, karena berbagai keterbatasan kami, Modul praktikum ini masih memiliki sejumlah kekurangan. Sehubungan dengan hal tersebut, kami mengharapkan masukan dari semua pihak, untuk perbaikan lebih lanjut.

Makassar, 09 Januari 2021

Penyusun

Wahyudi

PANDUAN UMUM

1. Modul ini adalah modul pendamping dari Trainer Robotika Berbasis Mikrokontroler.
2. Modul ini berisi 1 Materi dasar dan 8 percobaan yang berisi pembahasan teori aplikasi dan cara pemasangan aplikasi yang akan digunakan pada percobaan nantinya.
3. Modul ini berisi 8 Percobaan dimana terdiri dari:
   - 1 Materi dasar mikrokontroler AVR Atmega16 DIP sebagai pengantar dan dasar pengontrolan port;
   - 5 Percobaan menggunakan boar Arduino UNO dengan chip Atmega328 DIP sebagai pembelajaran dasar robotika dalam kontrol perangkat antarmuka;
   - 2 Percobaan menggunakan boar Arduino Nano dengan chip Atmega328 SMD sebagai pembelajaran dasar robotika dalam kontrol perangkat antarmuka;
   - 1 Percobaan menggunakan board nodeMCU dengan chip ESP8266 sebagai pembelajaran pengunaan kontrol jarak jauh robot atau Internet of Things (IoT)
4. Trainer telah dilengkapi 2 Arduino dan 1 nodemCU untuk dapat mendukung dan menyelesaikan 8 percobaan yang ada.
5. Trainer telah dilengkapi 4 perangkat atau komponen antarmuka tambahan untuk dapat mendukung dan menyelesaikan percobaan yang ada.
6. Trainer telah dilengkapi socket dan jumper yang mudah untuk mendukung perangkat antarmuka tambahan seperti modul , komponen dan lain sebagainya
7. Trainer telah dilengkapi dengan Socket Power VCC 12v, 5v dan GND untuk mendukung perangkat tambahan.
8. Trainer beroperasi menggunakan suplay tengangan utama penggunaan yang lebih maksimal menggunakan suplay utama dari adaptor/power supply dengan tegangan maksimal 12V.
9. Aplikasi dan Contoh Program terdapat pada CD dan dapat didownload dengan Scan Code berikut:

10. Pertanyaan dan saran dapat dikirim via email : wahyudi.elektronika@gamil.com atau @robotikngampus.id

DAFTAR ISI

DAFTAR GAMBAR

MATERI DASAR

ROBOTIKA DENGAN MIKROKONTROLLER

ROBOTIKA DENGAN MIKROKONTROLLER

A. Tentang Robotika

Kata robot berasal dari bahasa Cek yaitu robota, yang berarti pekerja. menurut arti bahasa, robot adalah sebuah alat mekanik yang dapat melakukan tugas _sik, baik menggunakan pengawasan dan kontrol manusia, atau menggunakan program yang telah di de_nisikan terlebih dahulu. Sedangkan menurut kamus besar bahasa indonesia, robot adalah alat berupa orang-orangan dan sebagainya dan dapat bergerak atau berbuat seperti manusia, dan dikendalikan oleh mesin, dengan berbagai macam bentuk dan jenis.

Gambar 1. Robot
*(**Sumber**: Robot Niryo One)*

Secara umum, ada dua jenis robot yaitu robot terkontrol *(controlled robot)* dan robot otomatis *(autonomous robot)*. Dalam kehidupan sehari-hari sering dijumpai mobil-mobilan yang dikendalikan dengan remote control. mainan ini merupakan salah satu contoh jenis *controled robot* yang sederhana. Mobil-mobilan tersebut akan bergerak maju atau mundur sesuai dengan perintah remote control yang dipegang oleh operator. jenis robot yang mampu mengambil keputusan sendiri adalah jenis robot otomatis *(autonomous robot)*. jenis robot ini tidak memerlukan kendali manusia untuk mengeksekusi sebuah keputusan. Ada banyak jenis robot yaitu robot manipulator, robot mobile, robot terbang, robot underwater dan robot humanoid.

1. Robot manipulator atau robot lengan diartikan sebagai sistem mekanik yang menunjukkan pergerakan dari **robot**. Sistem mekanik ini terdiri dari susunan link(rangka) dan joint (engsel) yang mampu menghasilkan gerakan yang terkontrol. **Manipulator** adalah bagian mekanik yang dapat

difungsikan untuk memindah, mengangkat, dan memanipulasi benda kerja.

2. Robot Mobile atau robot beroda adalah konstruksi **robot** yang ciri khasnya adalah mempunyai aktuator berupa roda untuk menggerakkan keseluruhan badan **robot** tersebut, sehingga **robot** tersebut dapat melakukan perpindahan posisi dari satu titik ke titik yang lain.
3. Robot Terbang adalah wahana **terbang** yang tidak berawak dapat dikendalikan oleh operator atau pilot dari jarak jauh ataupun oleh komputer yang ada didalamnya atau yang biasa disebut auto-pilot.
4. **Robot** Bawah Air atau Wahana Bawah Air Mandiri (Autonomous **Underwater** Vehicle, AUV) adalah **robot** yang di peruntukkan untuk mengamati obyek atau proses yg terjadi dalam air atau di dasar perairan.
5. Robot humanoid adalah robot yang penampilan keseluruhannya dibentuk berdasarkan tubuh manusia, mampu melakukan interaksi dengan peralatan maupun lingkungan yang dibuat untuk manusia.

B. Tentang Mikrokontroller

Mikrokontroler merupakan *chip* mikrokomputer yang secara fisik berupa sebuah IC (*Integrated Circuit*). Mikrokontroler biasanya digunakan dalam sistem kecil, murah dan tidak membutuhkan perhitungan yang sangat kompleks seperti dalam aplikasi di PC. Mikrokontroler banyak ditemukan dalam peralatan seperti microwave, oven, *keyboard*, CD *player*, VCR, *remote control*, robot, dll.

Gambar 2. Mikrokontroller
(***Sumber:*** *ndoware.com*)

Mikrokontroler berisikan bagian-bagian utama yaitu CPU (*Central Processing Unit*), RAM (*Random Access Memory*), ROM (*Read Only Memory*) dan *Port* I/O (*Input/Output*). Selain bagian-bagian utama tersebut, terdapat beberapa perangkat keras yang dapat digunakan untuk banyak keperluan seperti melakukan pencacahan, melakukan komunikasi serial, melakukan interupsi dll. mikrokontroler tertentu bahkan menyertakan ADC (*Analog to Digital Converter*), USB *Controller*, CAN (*Control Area Network*) dll.

Mikrokontroler bekerja berdasarkan program (perangkat lunak) yang ditanamkan didalamnya, dan program tersebut dibuat sesuai dengan aplikasi yang diinginkan. Aplikasi mikrokontroler normalnya terkait pembacaan data dari luar dan atau pengontrolan peralatan diluarnya. Contoh aplikasi yang sangat sederhana adalah melakukan pengendalian untuk menyalakan dan mematikan LED yang terhubung ke kaki mikrokontroler.

Mikrokontroler memiliki jalur-jalur masukan (*port* masukan) serta jalur-jalur keluaran (*port* keluaran) yang memungkinkan mikrokontroler tersebut untuk bisa digunakan dalam aplikasi pembacaan data, pengontrolan serta penyajian informasi. *Port* masukan digunakan untuk memasukkan informasi atau data dari luar ke mikrokontroler. Contoh informasi yang dimasukkan ke mikrokontroler ini adalah informasi kondisi saklar yang dihubungkan ke kaki mikrokontroler, apakah sedang terbuka atau tertutup. *Port* keluaran digunakan untuk mengeluarkan data atau informasi dari mikrokontroler. Adanya *port* keluaran ini memungkinkan mikrokontroler mengendalikan perangkat seperti LED, Motor, Relay dan menyajikan informasi melalui perangkat seperti Seven Segment dan LCD. Untuk bisa bekerja, mikrokontroler perlu diberikan tegangan dari luar. Umumnya IC mikrokontroler dapat bekerja pada tegangan 5V, namun demikian sebagian IC mikrokontroler juga dapat dioperasikan dengan tegangan 3V.

C. Tentang Arduino Uno

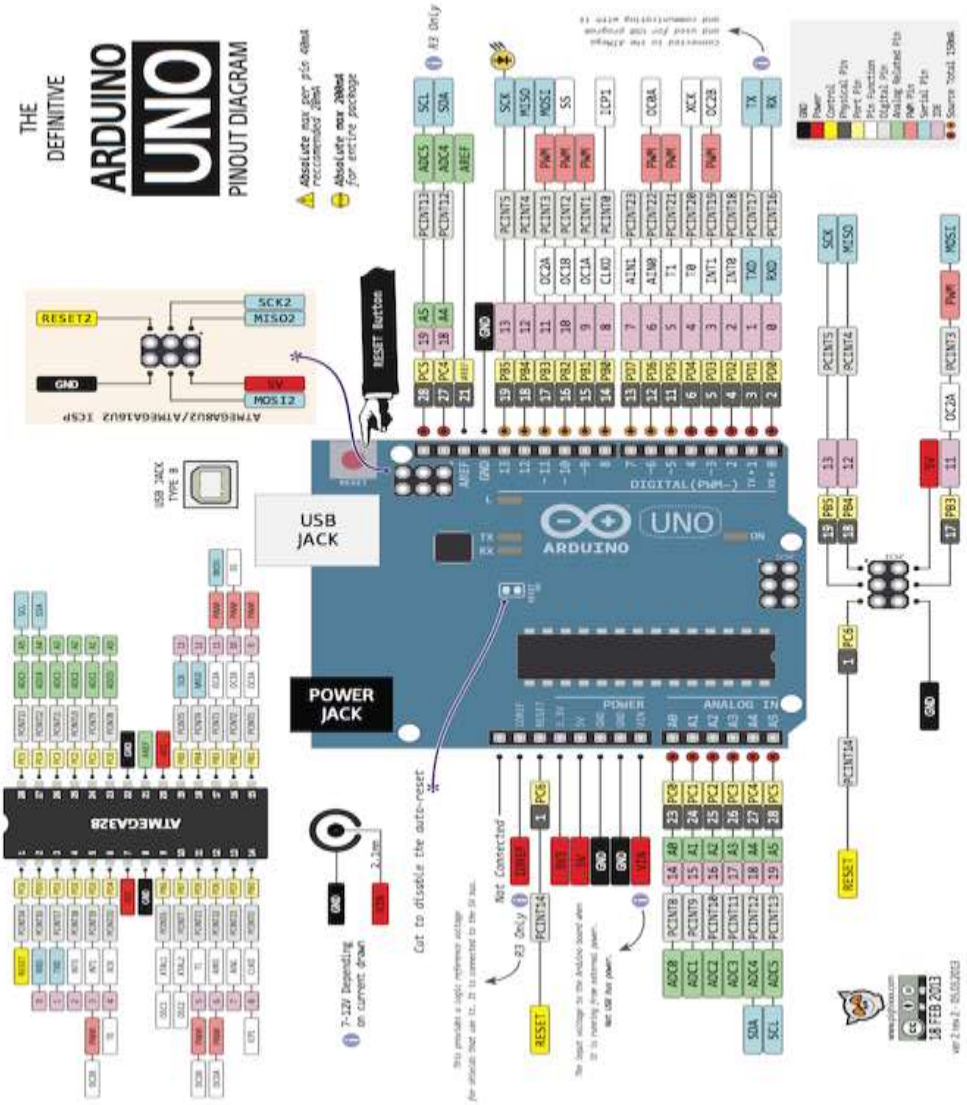
Arduino adalah sebuah board mikrokontroller yang berbasis ATmega328. Arduino memiliki 14 pin input/output yang mana 6 pin dapat digunakan sebagai output PWM, 6 analog input, crystal osilator 16 MHz, koneksi USB, jack power, kepala ICSP, dan tombol reset. Arduino mampu men-support

mikrokontroller; dapat dikoneksikan dengan komputer menggunakan kabel USB.

Arduino sebuah board minimum system mikrokontroler yang bersifat open source. Didalam rangkaian board arduino terdapat mikrokontroler AVR seri ATMega 328 yang merupakan produk dari atmel.

Gambar 3. Arduino Uno
(Sumber:** Arduino.cc**)

Arduino memiliki kelebihan tersendiri disbanding board mikrokontroler yang lain selain bersifat open source, arduino juga mempunyai bahasa pemrogramanya sendiri yang berupa bahasa C. Selain itu dalam board arduino sendiri sudah terdapat loader yang berupa USB sehingga memudahkan kita ketika kita memprogram mikrokontroler didalam arduino. Sedangkan pada kebanyakan board mikrokontroler yang lain yang masih membutuhkan rangkaian loader terpisah untuk memasukkan program ketika kita memprogram mikrokontroler. Port USB tersebut selain untuk loader ketika memprogram, bisa juga difungsikan sebagai port komunikasi serial. Arduino menyediakan 20 pin I/O, yang terdiri dari 6 pin input analog dan 14 pin digital input/output. Untuk 6 pin analog sendiri bisa juga difungsikan sebagai output digital jika diperlukan output digital tambahan selain 14 pin yang sudah tersedia. Untuk mengubah pin analog menjadi digital cukup mengubah konfigurasi pin pada program. Dalam board kita bisa lihat pin digital diberi keterangan 0-13, jadi untuk menggunakan pin analog menjadi output digital, pin analog yang pada keterangan board 0-5 kita ubah menjadi pin 14-19. dengan kata lain pin analog 0-5 berfungsi juga sebagi pin output digital 14-16.



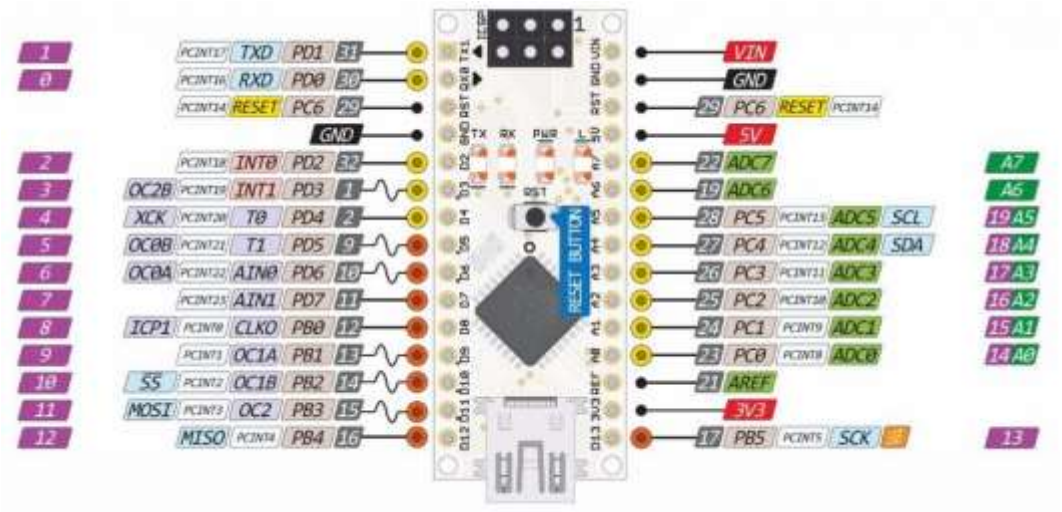
Gambar 4. Skema Arduino Uno
(**Sumber:** Arduino.cc)

D. Tentang Arduino Nano

Arduino merupakan sebuah platform dari physical computing yang bersifat open source. Arduino tidak hanya sekedar sebuah alat pengembang, tetapi merupakan kombinasi dari hardware, bahasa pemrogaman dan Integrated Development Environment (IDE) yang canggih IDE adalah sebuah software yang berperan untuk menulis program, meng-compile menjadi kode biner dan mengupload ke dalam memory mikrokontroler.

Gambar 5. Arduino Nano

Arduino Nano adalah salah satu board mikrokontroler yang berukuran kecil, lengkap dan mendukung penggunaan breadboard. Arduino Nano diciptakan dengan basis microcontroler ATmega328 (untuk Arduino Nano versi 3.x) atau Atmega 16(untuk Arduino versi 2.x). Arduino Nano kurang lebih memiliki fungsi yang sama dengan Arduino Duemilanove, tetapi dalam paket yang berbeda. ArduinoNano tidak menyertakan colokan DC berjenis Barrel Jack, dan dihubungkan ke komputer menggunakan port USB Mini-B. Arduino Nano dirancang dan diproduksi oleh perusahaan Gravitecth.



Gambar 6. Konfigurasi Arduino Nano

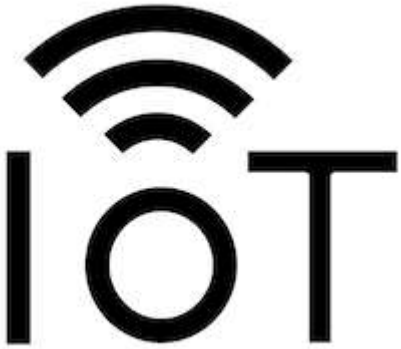
Arduino Nano dapat diaktifkan melalui koneksi USB Mini-B, atau melaluicatu daya eksternal dengan tegangan belum teregulasi antara 6-20 Volt yangdihubungkan melalui pin 30 atau pin VIN, atau melalui catu daya eksternaldengan tegangan teregulasi 5 volt melalui pin 27 atau pin 5V. Sumber

daya akansecara otomatis dipilih dari sumber tegangan yang lebih tinggi. Chip FTDIFT232L pada Arduino Nano akan aktif apabila memperoleh daya melalui USB,ketika Arduino Nano diberikan daya dari luar (Non-USB) maka Chip FTDI tidakaktif dan pin 3.3V pun tidak tersedia (tidak mengeluarkan tegangan), sedangkanLED TX dan RX pun berkedip apabila pin digital 0 dan 1 berada pada posisi HIGH.

E. Tentang NodeMCU ESP 8266

Internet of Things adalah suatu konsep dimana objek tertentu punya kemampuan untuk mentransfer data lewat jaringan tanpa memerlukan adanya interaksi dari manusia ke manusia ataupun dari manusia ke perangkat komputer.

Internet of Things leih sering disebut dengan singkatannya yaitu IoT. IoT ini sudah berkembang pesat mulai dari konvergensi teknologi nirkabel, micro-electromechanical systems (MEMS), dan juga Internet.IoT ini juga kerap diidentifikasikan dengan RFID sebagai metode komunikasi. Walaupun begitu, IoT juga bisa mencakup teknologi-teknologi sensor lainnya, semacam teknologi nirkabel maupun kode QR yang sering kita temukan di sekitar kita.



Gambar 7. IoT

Apa saja kemampuan dari IoT? Adapun kemampuannya bermacam-macam contohnya dalam berbagi data, menjadi remote control, dan masih banyak lagi yang lainnya. Sebenarnya fungsinya termasuk juga diterapkan ke benda yang ada di dunia nyata, di sekitar kita. Apa saja contohnya? Contohnya adalah untuk pengolahan bahan pangan, elektronik, dan berbagai mesin atau

teknologi lainnya yang semuanya tersambung ke jaringan lokal maupun global lewat sensor yang tertanam dan selalu menyala aktif.

Jadi, sederhananya istilah Internet of Things ini mengacu pada mesin atau alat yang bisa diidentifikasikan sebagai representasi virtual dalam strukturnya yang berbasis Internet. NodeMCU merupakan open source IOT platform yang ini termasuk firmware yang berjalan pada ESP8266 Wi-Fi SoC dari Espressif Systems, dan perangkat keras yang didasarkan pada modul ESP-12. Istilah "NodeMCU" secara default mengacu pada firmware daripada kit pengembangan.

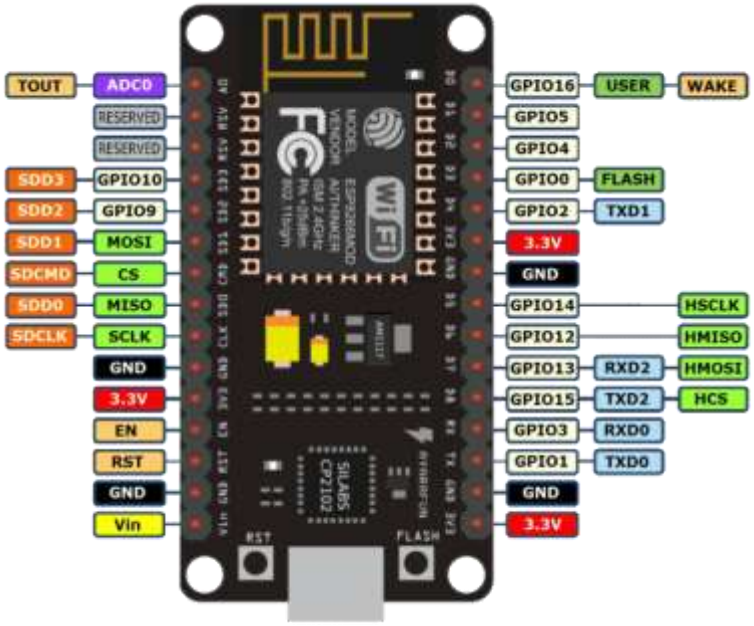
NodeMCU dibuat tidak lama setelah ESP8266 keluar. Pada 30 Desember 2013, Espressif Systems memulai produksi ESP8266. ESP8266 adalah SoC Wi-Fi yang terintegrasi dengan inti Tensilica Xtensa LX106, Banyak digunakan dalam aplikasi IoT. NodeMCU dimulai pada 13 Oktober 2014, ketika Hong melakukan file nodemcu-firmware pertama ke GitHub. Dua bulan kemudian, proyek ini diperluas untuk mencakup sebuah platform terbuka-hardware ketika pengembang Huang R berkomitmen dalam Gerber file papan ESP8266, bernama devkit v0.9. Kemudian pada bulan itu, Tuan PM mengirim pustaka klien MQTT dari Contiki ke platform SoP ESP8266, dan berkomitmen untuk proyek NodeMCU, kemudian NodeMCU dapat mendukung protokol MQTT IoT, menggunakan Lua untuk mengakses broker MQTT. Pembaruan penting lainnya dilakukan pada 30 Januari 2015, ketika Devsaurus memindahkan port u8glib ke proyek NodeMCU, memungkinkan NodeMCU untuk dengan mudah mengarahkan layar LCD, Layar, OLED, bahkan VGA.

Di musim panas 2015 para pencipta meninggalkan proyek firmware dan sekelompok kontributor independen mengambil alih. Pada musim panas 2016, NodeMCU menyertakan lebih dari 40 modul berbeda. Karena keterbatasan sumber daya, pengguna perlu memilih modul yang relevan untuk proyek mereka dan membuat firmware yang disesuaikan dengan kebutuhan mereka.

Berikut adalah spesifikasi, gambar, dan datasheet pin input output pada nodeMCU:

- Mikrokontroller. ESP8266.
- Ukuran Board. 57 mmx 30 mm.
- Tegangan Input. 3.3 ~ 5V.
- GPIO. 13 PIN.
- Kanal PWM. 10 Kanal.
- 10 bit ADC Pin. 1 Pin.
- Flash Memory. 4 MB.

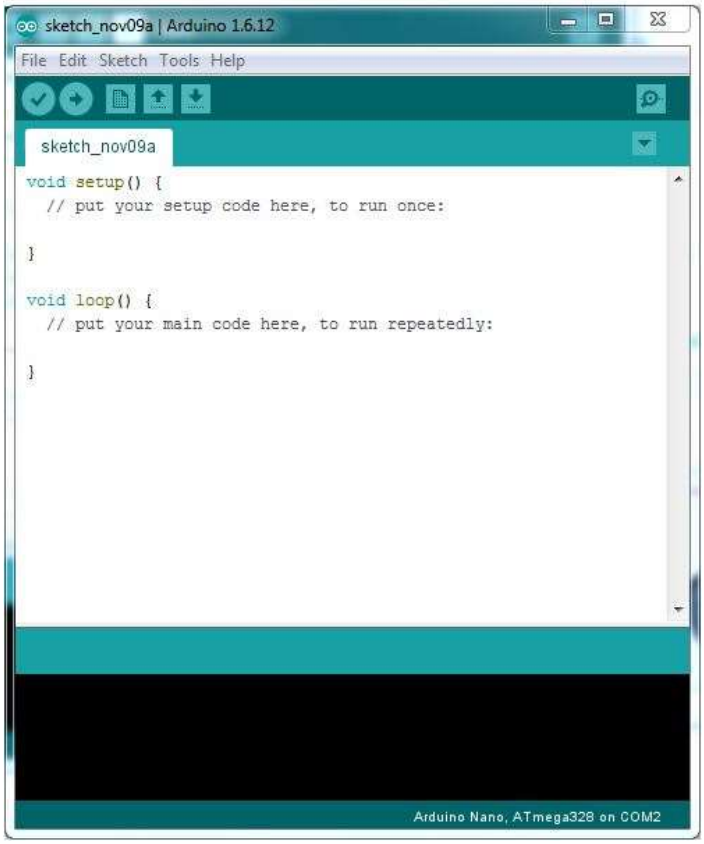
Data sheet :



Gambar 8. Datasheet NodeMCU

F. Tentang Arduino IDE

IDE merupakan akronim dari *Integrated Developtment Enviroenment,* atau secara istilah merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaksis pemrograman. Arduino IDE menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler.



Gambar 9. Tampilan aplikasi Arduino IDE

Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan *library* C/C++ yang biasa disebut *Wiring* yang membuat operasi *Input* dan *Output* menjadi lebih mudah. Arduino IDE ini dikembangkan dari *software processing* yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino.

Program yang ditulis dengan menggunakan Arduino *Software* (IDE) disebut sebagai *Sketch*. *Sketch* ditulis dalam suatu editor teks dan disimpan dalam *file* dengan ekstensi ***.ino***. Teks editor pada Arduino *Software* memiliki fitur" seperti *cutting/paste* dan *seraching/replacing* sehingga memudahkan dalam menulis kode program.

Pada *Software* Arduino IDE, terdapat semacam *message box* berwarna hitam yang berfungsi menampilkan status, seperti pesan *error*, *compile*, dan *upload* program. Di bagian bawah paling kanan *Software* Arduino IDE, menunjukkan *board* yang terkonfigurasi beserta COM *Port*s yang digunakan.

G. Variabel dan tipe data Pemrograman Arduino

1. Variabel

Variabel harus dipahami lebih dahulu sebelum membuat suatu program mikrokontroler, karena di dalam program mikrokontroler, banyak variabel yang bekerja di dalamnya. Variabel yang digunakan harus ditentukan juga tipe datanya. Variabel adalah nama yang dibuat dan disimpan di dalam memori mikrokontroler. Variabel ini mempunyai nilai dan nilainya dapat diubah sewaktu-waktu pada saat program dijalankan. Misalnya suatu variabel bernama "lampu1". Variabel ini merupakan suatu nilai yang memiliki tipe data tertentu. Nilai dalam variabel "lampu1" dapat diubah, misalnya variabel "lampu1" mula-mula bernilai 0, kemudian dalam suatu proses, variabel "lampu1" dapat berubah menjadi bernilai 1. Nilai 0 dan 1 adalah tipe data dalam variabel "lampu1".

Contoh lainnya, misalnya kita mendeklarasikan suatu variabel bernama "cacah", kemudian ditentukan tipe data adalah byte. Maka program akan membuat suatu variabel "cacah" dan memberikan tipe data "cacah" sebagai bilangan byte dan menyimpannya ke memori. Artinya " variabel "cacah" hanya dapat menghitung dari 0 hingga 255 (8 bit).

Deklarasi suatu variabel dapat dilakukan tanpa pemberian nilai awal atau dapat juga langsung diberikan nilai awal. Misalnya jika kita mendeklarasikan variabel "hitung" sebagai bilangan bulat (integer) dengan nilai awal 1 maka dapat ditulis:

```
boolean hitung = 1;
```

maka mikrokontroler akan membuat sebuah variabel "hitung" dengan tipe data integer (bilangan bulat) dan memberikan nilai awal 1 ke variabel "hitung" tersebut.

Dalam pemograman mikrokontroler dikenal ada 2 macam variabel yaitu:

- Variabel global; yaitu variabel yang dideklarasi diluar fungsi dan berlaku secara umum atau dapat diakses dimana saja.
- Variabel lokal; yaitu variabel yang dideklarasi di dalam fungsi dan hanya dapat diakses oleh pernyataan yang ada di dalam fungsi.

2. Tipe Data

Tipe data yang dapat digunakan di dalam program *Sketch* bermacam-macam antara lain:

a. Boolean

Tipe data boolean hanya memiliki 2 data yaitu benar (*true*) dan salah (*false*). Tipe data boolean hanya membutuhkan 1byte memori. Contoh penulisan deklarasi tipe data boolean.

boolean running = false;

Artinya: variabel "running" ditentukan mempunyai tipe data boolean dengan nilai awal "false".

b. Byte

Tipe data byte memiliki 8 bit data (0 – 255). Tipe data byte tidak memiliki nilai negatif. Contoh penulisan deklarasi tipe data byte:

byte b = 128;

atau

byte b = B10000000; Artinya variabel "b" ditentukan sebagai tipe daya byte dengan nilai awal 128 (decan) atau B10000000.

c. Char

Char Char atau karakter adalah tipe data untuk menyatakan suatu karakter seperti "A", "B", "C", "+", "." dan sebagainya. Karakter ini disimpan dalam bentuk angka. Untuk mengonversikan bentuk angka ke bentuk karakter dapat menggunakan tabel ASCII.

Decimal 128 64 32 16 8 4 2 1

Binary 0 1 1 1 1 1 1 1 = 127 Decimal

Contoh penulisan deklarasi variabel tipe char:

char karakterku = 68;

Artinya mendeklarasikan suatu variabel dengan nama "karakterku" dengan nilai awal 68 atau karakter "D".

d. unsigned char

unsigned char serupa dengan tipe char tetapi tanpa nilai negatif sehingga unsigned char memiliki nilai dari 0 hingga 255 atau sama dengan tipe data byte.

Contoh penulisan deklarasi variabel tipe unsigned char:

unsigned char bilangan = -1;

Artinya mendeklarasikan suatu variabel dengan nama "bilangan" dengan nilai awal -1.

e. Int

int digunakan untuk menyatakan tipe data integer (bilangan bulat). Tipe data integer berkisar antara -32768 sampai 32768 (-215 hingga (215-1)). Nilai bilangan ini membutuhkan 2 byte memori mikrokontroler. Contoh penulisan deklarasi tipe integer

int bilangan = 0;

Artinya mendeklarasikan suatu variabel dengan nama "bilangan" dengan nilai awal 0.

f. Unsigned int

Unsigned int digunakan untuk mendeklarasikan tipe data bilangan bulat positif atau bernilai 0 hingga 65535. Sama dengan integer, tipe data ini juga membutuhkan 2 byte memori mikrokontroler. Contoh penulisannya:

unsigned int angka = 12;

Artinya mendeklarasikan suatu variabel dengan nama "angka" dengan nilai awal 12.

g. word

Tipe data word sama dengan tipe data unsigned int. Besar data word adalah 16 bit atau membutuhkan 2 byte memori mikrokontroler. Tipe data ini jarang digunakan. Contoh penulisan pada program *Sketch*:

word bilangan = 1000;

Artinya mendeklarasikan variabel dengan nama "angka" dengan nilai awal 1000.

h. long

Tipe data long adalah tipe data untuk menampung bilangan bulat yang berkisar antara -2.147.483.648 hingga 2.147.483.647. variabel bertipe long ditulis dengan akhiran L atau l. Contoh 456789980L. Contoh penulisan pada program *Sketch*:

long speedoflight = 298756L

artinya program akan mendeklarasikan variabel dengan nama“speedoflight” dengan nilai 298756.

i. unsigned long

unsigned long adalah tipe data yang sama dengan long, tetapi dihitung dari angka 0 atau mempunyai nilai berkisar 0 hingga 4.292.967.295. Nilai variabel ini ditulis dengan kode UL di akhir konstanta seperti contoh berikut ini

unsigned long speed = 123345623UL

artinya program mendeklarasikan variabel dengan nama “speed” dengan nilai awal 123345623.

j. float

Tipe ini berguna untuk menyimpan bilangan real. Angka yang bisa disimpan dari -3,4028235 x 1038 hingga 3,4028235 x 1038. Angka dengan tipe float sangat besar sekali, sehingga sangat jarang digunakan karena akan memperlambat kerja prosesor mikrokontroler dan banyak memakai memori, kecuali bila memang sangat dibutuhkan dalam program.

k. double

Tipe data double dan float untuk arduino tidak ada bedanya

l. Konstanta

Konstanta adalah nilai suatu besaran yang tidak berubah besarnya atau memiliki nilai yang tetap. Dalam pemograman arduino ditulis dengan kata “const” yang artinya konstanta. Contoh penulisan konstanta :

const int harga = 12;

artinya variabel “harga” adalah sebuah konstanta bilangan bulat integer yang memiliki nilai tetap yaitu 12. Nilai ini hanya bisa dibaca dan tidak dapat diubah selama program dijalankan.

Penamaan untuk variabel dan konstanta tidak boleh menggunakan kata-kata yang digunakan dalam program *Sketch* arduino seperti for, if, while dan sebagainya. Selain itu nama tidak boleh ada spasi, bila hendak memisahkan 2 buah kata untuk menyatakan variabel atau konstanta harus dihubungkan dengan tanda garis bawah (_) misalnya: “waktu_tunggu”.

PERCOBAAN I

ANTARMUKA ROBOT MANIPULATOR

PERCOBAAN I | ANTARMUKA ROBOT MANIPULATOR

A. Tujuan Percobaan

1. Mahasiswa mampu menjelaskan prinsip dasar motor servo .
2. Mahasiswa mampu mengimplementasikan motor servo sebagai robot manipulator/robot lengan pada trainer robotika.
3. Mahasiswa mampu melakukan pemograman motor servo pada aplikasi arduino IDE sebagai robot manipulator/robot lengan.

B. Teori Dasar

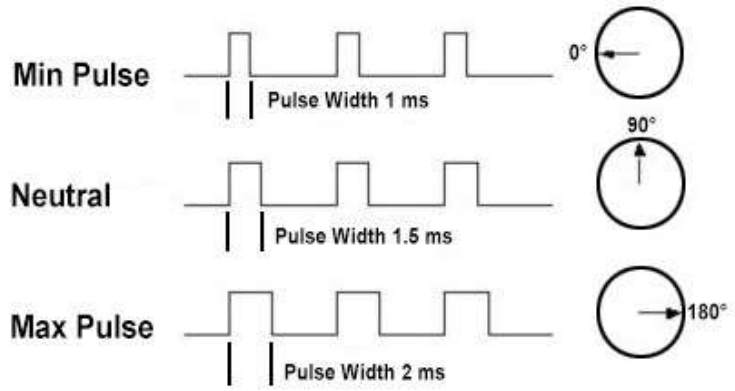
Motor servo merupakan motor listrik dengan sistem *closed loop* yang digunakan untuk mengendalikan kecepatan, akselerasi dan posisi akhir dari sebuah motor listrik dengan keakuratan yang tinggi. Motor servo terdiri dari tiga bagian utama, yaitu: motor, sistem kontrol dan potensiometer/encoder yang terhubung dengan satu set roda gigi ke poros *Output*. Potensiometer atau encoder ini lah yang berfungsi sebagai sensor yang memberikan sinyal umpan balik (feedback) ke sistem kontrol apakah posisi targetnya sudah benar atau belum. Encoder biasanya digunakan pada motor servo industri. Sedangkan Potensiometer biasanya digunakan pada aplikasi yang lebih sederhana seperti mobil *remote control*. Potensiometer ini terdiri dari tiga kabel dengan 2 kabel untuk power dan 1 kabel untuk kabel sinyal.

Gambar 10. Motor Servo

Motor akan menggerakkan roda gigi untuk memutar potensiometer dan poros *Output* secara bersamaan. Potensiometer yang akan mengendalikan posisi sudut motor servo dengan pemberian sinyal ke dalam sistem kontrol. Jika posisi targetnya sudah benar, maka ia akan menghentikan motor servo. Sebaliknya, Jika sistem kontrol mendeteksi bahwa sudut belum tepat, maka ia

akan mengubah motor servo ke arah yang benar sampai posisi sudutnya benar. Kelebihan inilah yang tidak ditemukan pada motor biasa. Motor servo biasanya digunakan untuk mengendalikan posisi sudut antara 0 dan 180 derajat.

Motor servo dikendalikan dengan sinyal PWM dari encoder/Potensiometer. Lebar sinyal (pulsa) yang diberikan inilah yang akan menentukan posisi sudut putaran dari poros motor servo. Sebagai contoh, lebar sinyal dengan waktu 1,5 ms (mili second) akan memutar poros motor servo ke posisi sudut 90º. Bila sinyal lebih pendek dari 1,5 ms maka akan berputar ke arah posisi 0º atau ke kiri (berlawanan dengan arah jarum jam), sedangkan bila sinyal yang diberikan lebih lama dari 1,5 ms maka poros motor servo akan berputar ke arah posisi 180º atau ke kanan (searah jarum jam). Lebih jelasnya perhatikan gambar berikut ini:

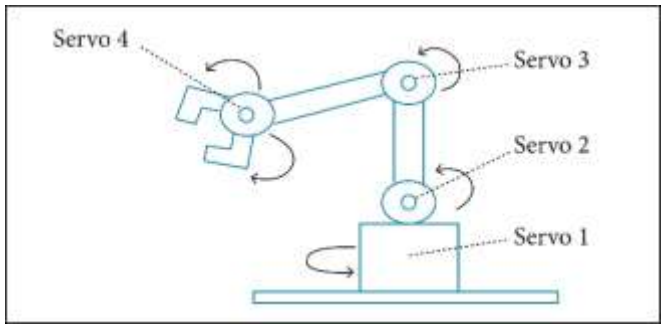


Gambar 11. Pengontrolan Motor Servo dengan sinyal PWM (***Sumber:*** *www.insinyoer.com*)

Ketika sinyal PWM telah diberikan, maka poros motor servo akan bergerak ke posisi yang telah ditargetkan dan berhenti pada posisi tersebut serta akan tetap bertahan pada posisi tersebut. Jika ada kekuatan eksternal yang mencoba memutar atau mengubah posisi tersebut, maka sistem closed loop dari motor servo tersebut akan bekerja dengan mencoba menahan atau melawan kekuatan eksternal tersebut dengan kekuatan internal dari motor servo itu sendiri. Namun motor servo tidak akan mempertahankan posisinya untuk selamanya, sinyal PWM harus diulang setiap 20 ms (mili second) agar posisi poros motor servo tetap bertahan pada posisinya. Berikut ini adalah ilustrasi dari perbedaan *open loop system vs closed loop system* dimana motor

servo mengandalkan closed loop system dengan sinyal umpan balik (feedback) sehingga posisi yang ditargetkan akan tergapai secara otomatis.

Teknologi Robot Manipulator yang berbentuk mirip dengan lengan manusia dan berfungsi membantu pekerjaan manusia yaitu dengan mengaplikasikan lengan robot tersebut didalam dunia industri, dapat digunakan sebagai pemindah barang dengan berat barang berskala besar, dengan kecepatan dan ketepatan yang akurat, pengendaliannya pun bisa berupa otomatis atau secara manual. Otomatis robot merupakan robot yang dapat bergerak sesuai dengan sistem geraknya tanpa harus ada campur tangan manusia. Manual robot merupakan robot yang bergerak sesuai dengan sistem geraknya tapi dengan bantuan operator sebagai pengendalinya.



Gambar 12. 4DOF

Lengan robot pada umumnya terdiri dari bahu, persendian dan tangan yang bisa berupa sebuah gripper atau tangan yang memiliki jari seperti halnya tangan manusia sebagai pengambil objek. Bagian tangan robot dikenal sebagai manipulator tangan, yaitu sistem gerak yang berfungsi untuk manipulasi (memegang, mengambil, mengangkat, memindahkan, mengolah) objek. Untuk melakukan pengambilan objek lengan robot ini dilengkapi dengan gripper (pemegang) yang berupa jari-jari seperti halnya jari manusia. Lengan robot didesain agar dapat mengikuti gerak sesuai dengan gerakan yang dilakukan oleh gerakan lengan manusia, input pengontrol dibuat dengan potensiometer untuk persendian lengan dan Flex Sensor yang diletakkan pada jari-jari manusia dengan cara membuat pengendali yang sesuai dengan bentuk lengan dan jari-jari manusia agar dapat digunakan sebagai penggerak sendi-sendi pada lengan robot.

Gambar 13. Robot Manipulator 4DOF

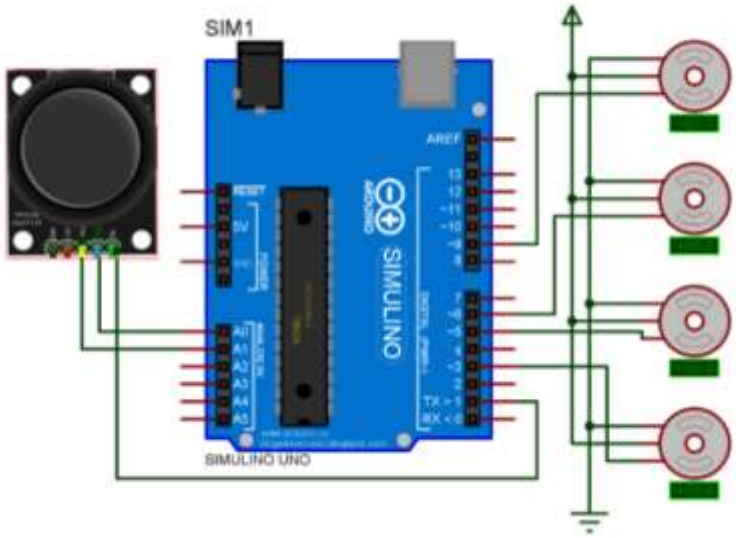
Manipulator merupakan sistem mekanik yang menunjukkan pergerakan dari robot. Sistem mekanik ini terdiri dari susunan link (rangka) dan joint (engsel) yang mampu menghasilkan gerakan yang terkontrol, sebagai rangakaian umpan balik terbuka maupun rangkaian umpan balik tertutup yang dihubungkan dengan sendi-sendi dan dapat melakukan gerakan-gerakan secara bebas.

C. Alat dan Bahan

1. Trainer Robotika 1 buah
2. Komputer/laptop 1 buah
3. Kabel USB Arduino 1 buah
4. Jumper Male-male 9 buah

D. Langkah Percobaan

1. Buatlah rangkaian robot manipulator dengan mengikuti skema berikut :

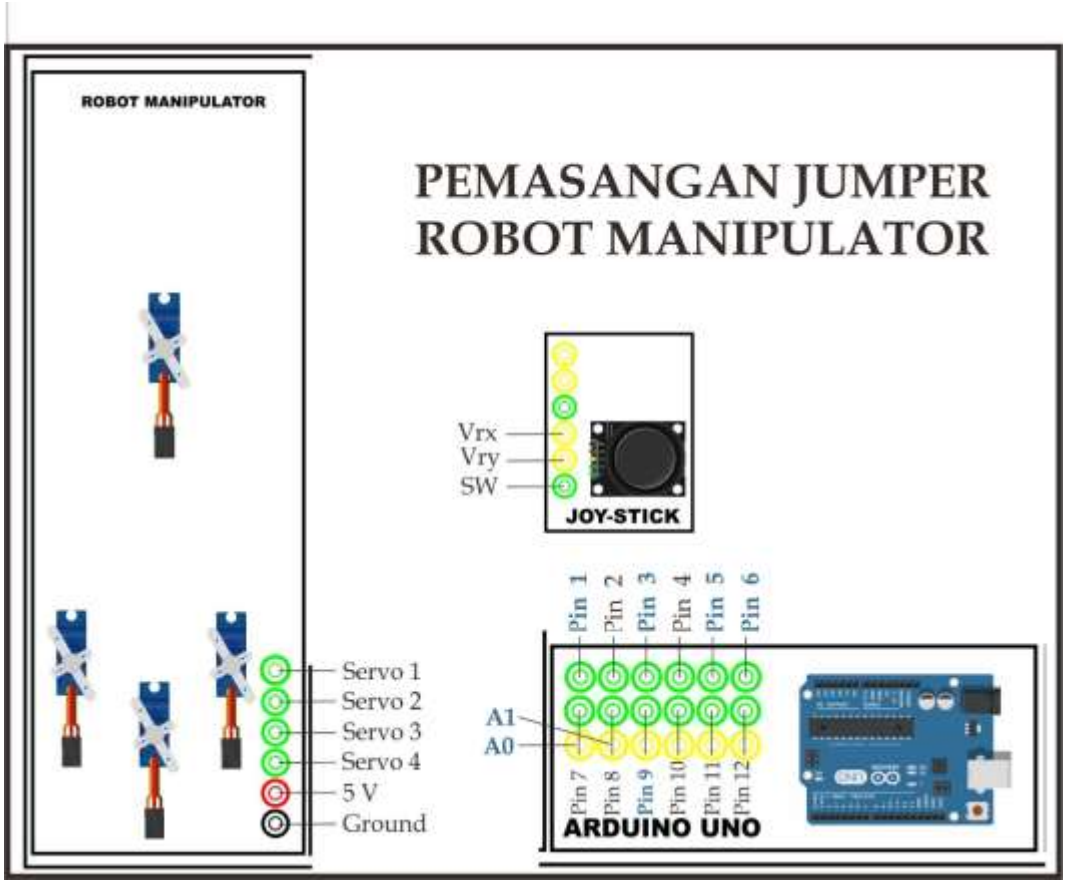


Gambar 14. Skema rangkaian robot manipulator

Tabel 2.1. Konfigurasi pin rangkaian robot manipulator

Arduino Uno	**Antarmuka**
Pin 3	Servo 1
Pin 5	Servo 2
Pin 6	Servo 3
Pin 9	Servo 4
Pin 1	SW
A0	VRx
A1	VRy
	Red to 5V
	Black to GND

Rangkaian tersebut dapat dipraktikkan pada trainer dengan menghubungkan pin 3 Arduino Uno ke Servo 1, pin 5 Arduino Uno ke Servo 2, pin 6 Arduino Uno ke Servo 3, pin 9 Arduino Uno ke Servo 4 , pin 1 Arduino Uno ke SW Joystick, A0 Arduino Uno ke VRx Jostick, A1 Arduino Uno ke VRy Joystick kemudian Red to 5V , dan terakhir Black to Ground, seperti gambar pemasangan *jumper* pada trainer berikut ini:



Gambar 15. Rangkaian Robot manipulator pada trainer

2. Buka aplikasi Arduino IDE pada komputer/laptop dan masukkan *Sketch* program berikut :

```
#include <Servo.h>

bool repeatePlaying = false; /* Repeatedly is running recorded
cycle */
int delayBetweenCycles = 2000; /* Delay between cycles */

int basePin = 3;        /* Base servo */
int shoulderPin = 5;    /* Shoulder servo */
int elbowPin = 6;        /* Elbow servo */
int gripperPin = 9;      /* Gripper servo */

int xdirPin = A0;         /* Base - joystick1*/
int ydirPin = A1;         /* Shoulder - joystick1 */
int zdirPin = 1;         /* Elbow - joystick2 */
int gdirPin = 2;         /* Gripper - joystick2 */

int pinRecord = PD2;      /* Button record - recommended (A4 is
deprecated, will by used for additional joystick) */
int pinPlay = PD3;        /* Button play  - recommended (A5 is
deprecated, will by used for additional joystick) */
int pinLedRecord = PD4;  /* LED - indicates recording (light)
or auto play mode (blink one) */
```

```
bool useInternalPullUpResistors = false;

const int buffSize = 512; /* Size of recording buffer */

int startBase = 90;
int startShoulder = 90;
int startElbow = 90;
int startGripper = 0;

int posBase = 90;
int posShoulder = 90;
int posElbow = 90;
int posGripper = 0;

int lastBase = 90;
int lastShoulder = 90;
int lastElbow = 90;
int lastGripper = 90;

int minBase = 0;
int maxBase = 150;
int minShoulder = 0;
int maxShoulder = 150;
int minElbow = 0;
int maxElbow = 150;
int minGripper = 0;
int maxGripper = 150;

const int countServo = 4;
int buff[buffSize];
int buffAdd[countServo];
int recPos = 0;
int playPos = 0;

int buttonRecord = HIGH;
int buttonPlay = HIGH;

int buttonRecordLast = LOW;
int buttonPlayLast = LOW;

bool record = false;
bool play = false;
bool debug = false;

String command = "Manual";
int printPos = 0;

int buttonPlayDelay = 20;
int buttonPlayCount = 0;

bool ledLight = false;

Servo servoBase;
Servo servoShoulder;
Servo servoElbow;
Servo servoGripper;

void setup() {
  Serial.begin(9600);
```

```
  if (useInternalPullUpResistors) {
    pinMode(pinRecord, INPUT_PULLUP);
    pinMode(pinPlay, INPUT_PULLUP);
  }
  else
  {
    pinMode(pinRecord, INPUT);
    pinMode(pinPlay, INPUT);
  }

  pinMode(xdirPin, INPUT);
  pinMode(ydirPin, INPUT);
  pinMode(zdirPin, INPUT);
  pinMode(gdirPin, INPUT);

  pinMode(pinLedRecord, OUTPUT);

  servoBase.attach(basePin);
  servoShoulder.attach(shoulderPin);
  servoElbow.attach(elbowPin);
  servoGripper.attach(gripperPin);

  StartPosition();

  digitalWrite(pinLedRecord, HIGH);
  delay(1000);
  digitalWrite(pinLedRecord, LOW);
}

void loop() {

  buttonRecord = digitalRead(pinRecord);
  buttonPlay = digitalRead(pinPlay);

  //  Serial.print(buttonRecord);
  //  Serial.print("\t");
  //  Serial.println(buttonPlay);
  //  for testing purposes

  if (buttonPlay == LOW)
  {
    buttonPlayCount++;

    if (buttonPlayCount >= buttonPlayDelay)
    {
      repeatePlaying = true;
    }
  }
  else buttonPlayCount = 0;

  if (buttonPlay != buttonPlayLast)
  {
    if (record)
    {
      record = false;
    }

    if (buttonPlay == LOW)
```

```
    {
      play = !play;
      repeatePlaying = false;

      if (play)
      {
        StartPosition();
      }
    }
  }

  if (buttonRecord != buttonRecordLast)
  {
    if (buttonRecord == LOW)
    {
      record = !record;

      if (record)
      {
        play = false;
        repeatePlaying = false;
        recPos = 0;
      }
      else
      {
        if (debug) PrintBuffer();
      }
    }
  }

  buttonPlayLast = buttonPlay;
  buttonRecordLast = buttonRecord;

  float dx = map(analogRead(xdirPin), 0, 1023, -5.0, 5.0);
  float dy = map(analogRead(ydirPin), 0, 1023, 5.0, -5.0);
  float dz = map(analogRead(zdirPin), 0, 1023, 5.0, -5.0);
  float dg = map(analogRead(gdirPin), 0, 1023, 5.0, -5.0);

  if (abs(dx) < 1.5) dx = 0;
  if (abs(dy) < 1.5) dy = 0;
  if (abs(dz) < 1.5) dz = 0;
  if (abs(dg) < 1.5) dg = 0;

  posBase += dx;
  posShoulder += dy;
  posElbow += dz;
  posGripper += dg;

  if (play)
  {
    if (playPos >= recPos) {
      playPos = 0;

      if (repeatePlaying)
      {
        delay(delayBetweenCycles);
        StartPosition();
      }
      else
```

```
    {
      play = false;
    }
  }

  bool endOfData = false;

  while (!endOfData)
  {
    if (playPos >= buffSize - 1) break;
    if (playPos >= recPos) break;

    int data = buff[playPos];
    int angle = data & 0xFFF;
    int servoNumber = data & 0x3000;
    endOfData = data & 0x4000;

    switch (servoNumber)
    {
      case 0x0000:
        posBase = angle;
        break;

      case 0x1000:
        posShoulder = angle;
        break;

      case 0x2000:
        posElbow = angle;
        break;

      case 0x3000:
        posGripper = angle;
        dg = posGripper - lastGripper;
        break;
    }

    playPos++;
  }
}

if (posBase > maxBase) posBase = maxBase;
if (posShoulder > maxShoulder) posShoulder = maxShoulder;
if (posElbow > maxElbow) posElbow = maxElbow;
if (posGripper > maxGripper) posGripper = maxGripper;

if (posBase < minBase) posBase = minBase;
if (posShoulder < minShoulder) posShoulder = minShoulder;
if (posElbow < minElbow) posElbow = minElbow;
if (posGripper < minGripper) posGripper = minGripper;

servoBase.write(posBase);
servoShoulder.write(posShoulder);
servoElbow.write(posElbow);

bool waitGripper = false;
if (dg < 0) {
  posGripper = minGripper;
  waitGripper = true;
```

```
  }
  else if (dg > 0) {
    posGripper = maxGripper;
    waitGripper = true;
  }

  servoGripper.write(posGripper);
  if (play && waitGripper)
  {
    delay(1000);
  }

  if ((lastBase != posBase) | (lastShoulder != posShoulder) |
(lastElbow != posElbow) | (lastGripper != posGripper))
  {
    if (record)
    {
      if (recPos < buffSize - countServo)
      {
        int buffPos = 0;

        if (lastBase != posBase)
        {
          buffAdd[buffPos] = posBase;
          buffPos++;
        }

        if (lastShoulder != posShoulder)
        {
          buffAdd[buffPos] = posShoulder | 0x1000;
          buffPos++;
        }

        if (lastElbow != posElbow)
        {
          buffAdd[buffPos] = posElbow | 0x2000;
          buffPos++;
        }

        if (lastGripper != posGripper)
        {
          buffAdd[buffPos] = posGripper | 0x3000;
          buffPos++;
        }

        buffAdd[buffPos - 1] = buffAdd[buffPos - 1] | 0x4000;

        for (int i = 0; i < buffPos; i++)
        {
          buff[recPos + i] = buffAdd[i];
        }

        recPos += buffPos;
      }
    }

    command = "Manual";
    printPos = 0;
```

```
    if (play)
    {
      command = "Play";
      printPos = playPos;
    }
    else if (record)
    {
      command = "Record";
      printPos = recPos;
    }

    Serial.print(command);
    Serial.print("\t");
    Serial.print(printPos);
    Serial.print("\t");
    Serial.print(posBase);
    Serial.print("\t");
    Serial.print(posShoulder);
    Serial.print("\t");
    Serial.print(posElbow);
    Serial.print("\t");
    Serial.print(posGripper);
    Serial.print("\t");
    Serial.print(record);
    Serial.print("\t");
    Serial.print(play);
    Serial.println();
  }

  lastBase = posBase;
  lastShoulder = posShoulder;
  lastElbow = posElbow;
  lastGripper = posGripper;

  if ( repeatePlaying)
  {
    ledLight = !ledLight;
  }
  else
  {
    if (ledLight)
    {
      ledLight = false;
    }

    if (record)
    {
      ledLight = true;
    }
  };

  digitalWrite(pinLedRecord, ledLight);
  delay(50);
}

void PrintBuffer()
{
  for (int i = 0; i < recPos; i++)
  {
```

```
      int data = buff[i];
      int angle = data & 0xFFF;
      int servoNumber = data & 0x3000;
      bool endOfData = data & 0x4000;

      Serial.print("Servo=");
      Serial.print(servoNumber);
      Serial.print("\tAngle=");
      Serial.print(angle);
      Serial.print("\tEnd=");
      Serial.print(endOfData);
      Serial.print("\tData=");
      Serial.print(data, BIN);
      Serial.println();
    }
}

void StartPosition()
{
  int angleBase = servoBase.read();
  int angleShoulder = servoShoulder.read();
  int angleElbow = servoElbow.read();
  int angleGripper = servoGripper.read();

  Serial.print(angleBase);
  Serial.print("\t");
  Serial.print(angleShoulder);
  Serial.print("\t");
  Serial.print(angleElbow);
  Serial.print("\t");
  Serial.print(angleGripper);
  Serial.println("\t");

  posBase = startBase;
  posShoulder = startShoulder;
  posElbow = startElbow;
  posGripper = startGripper;

  servoBase.write(posBase);
  servoShoulder.write(posShoulder);
  servoElbow.write(posElbow);
  servoGripper.write(posGripper);
}
```

3. Compile program dengan menekan tombol *Verify/Compile* pada aplikasi Arduino IDE atau dengan menekan tombol Ctrl + R pada *keyboard* untuk memastikan program berhasil dan tidak terdapat kesalahan.
4. Hubungkan Laptop dengan Arduino UNO menggunakan Kabel USB Arduino
5. Upload program pada computer/laptop ke Arduino Uno dengan menekan tombol Upload pada aplikasi Arduino IDE atau dengan menekan tombol Ctrl + U pada *keyboard* dan pastikan proses upload selesai.

6. Buatlah analisis data dan kesimpulan

E. Hasil Percobaan

1. Kondisi awal robot manipulator/lengan dalam keadaan tidak bergerak;
2. Posisi Joystick ke atas (Y) ditekan maka Servo 2 dan 3 akan berotasi sehingga Robot Manipulator/ARM Robot Bergerak Kedepan;
3. Posisi Joystick ke samping (X) ditekan maka Servo 1 akan berotasi sesuai dengan derajat pada program sehingga Robot Manipulator/ARM Robot Bergerak kiri/kanan;

PERCOBAAN II

ANTARMUKA ROBOT MOBILE *(WALL FOLLOWER)*

PERCOBAAN II | ANTARMUKA ROBOT MOBILE *(WALL FOLLOWER)*

A. Tujuan Percobaan

1. Mahasiswa mampu menjelaskan prinsip kerja sensor ultrasonic.
2. Mahasiswa mampu mengimplementasikan sensor Ultrasonic sebagai robot *wall follower* pada rangkaian Trainer robotika.
3. Mahasiswa mampu melakukan pemograman sensor Ultrasonic sebagai robot *wall follower* pada Aplikasi Arduino IDE.

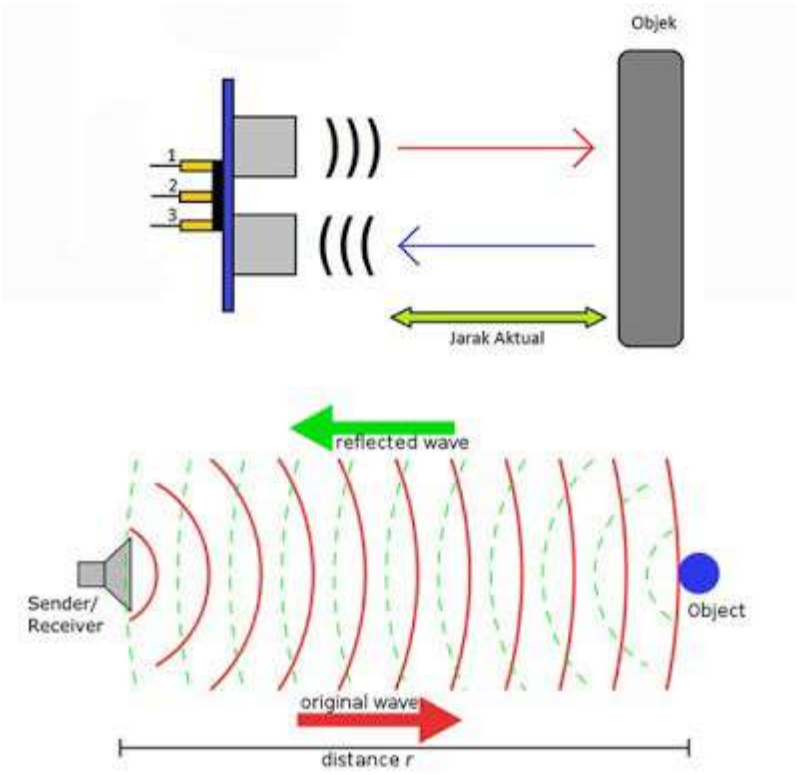
B. Teori Dasar

Sensor Ultrasonic adalah sebuah sensor yang berfungsi untuk mengubah besaran fisik (bunyi) menjadi besaran listrik dan sebaliknya. Cara kerja sensor ini didasarkan pada prinsip dari pantulan suatu gelombang suara sehingga dapat dipakai untuk menafsirkan eksistensi (jarak) suatu benda dengan frekuensi tertentu. Disebut sebagai sensor Ultrasonic karena sensor ini menggunakan gelombang Ultrasonic (bunyi Ultrasonic).

Gambar 16. Sensor Ultrasonic HC SR04

Gelombang Ultrasonic adalah gelombang bunyi yang mempunyai frekuensi sangat tinggi yaitu 20.000 Hz. Bunyi Ultrasonic tidak dapat di dengar oleh telinga manusia. Bunyi Ultrasonic dapat didengar oleh anjing, kucing, kelelawar, dan lumba-lumba. Bunyi Ultrasonic bisa merambat melalui zat padat, cair dan gas. Reflektivitas bunyi Ultrasonic di permukaan zat padat hampir sama dengan reflektivitas bunyi Ultrasonic di permukaan zat cair. Akan tetapi, gelombang bunyi Ultrasonic akan diserap oleh tekstil dan busa.

Pada sensor Ultrasonic, gelombang Ultrasonic dibangkitkan melalui sebuah alat yang disebut dengan piezoelektrik dengan frekuensi tertentu. Piezoelektrik ini akan menghasilkan gelombang Ultrasonic (umumnya berfrekuensi 40kHz) ketika sebuah osilator diterapkan pada benda tersebut. Secara umum, alat ini akan menembakkan gelombang Ultrasonic menuju suatu area atau suatu target. Setelah gelombang menyentuh permukaan target, maka target akan memantulkan kembali gelombang tersebut. Gelombang pantulan dari target akan ditangkap oleh sensor, kemudian sensor menghitung selisih antara waktu pengiriman gelombang dan waktu gelombang pantul diterima.



Gambar 17. Cara kerja sensor Ultrasonic

Secara detail, cara kerja sensor Ultrasonic adalah sebagai berikut:

1. Sinyal dipancarkan oleh pemancar Ultrasonic dengan frekuensi tertentu dan dengan durasi waktu tertentu. Sinyal tersebut berfrekuensi diatas 20kHz. Untuk mengukur jarak benda (sensor jarak), frekuensi yang umum digunakan adalah 40kHz.

2. Sinyal yang dipancarkan akan merambat sebagai gelombang bunyi dengan kecepatan sekitar 340 m/s. Ketika menumbuk suatu benda, maka sinyal tersebut akan dipantulkan oleh benda tersebut.
3. Setelah gelombang pantulan sampai di alat penerima, maka sinyal tersebut akan diproses untuk menghitung jarak benda tersebut. Jarak benda dihitung berdasarkan rumus:

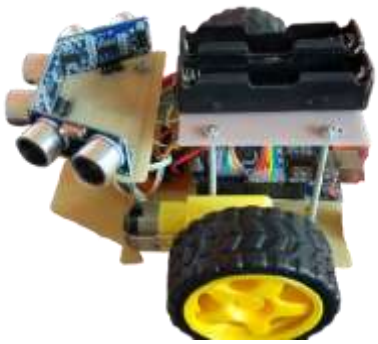
   S = 340.t/2

   dimana S merupakan jarak antara sensor Ultrasonic dengan benda (bidang pantul), dan t adalah selisih antara waktu pemancaran gelombang oleh transmitter dan waktu ketika gelombang pantul diterima receiver.

Fitur Sensor Ulrasonic HC-SR04:

- Tegangan operasi: + 5V
- Jarak Pengukuran Teoritis: 2cm hingga 450cm
- Jarak Pengukuran Praktis: 2cm hingga 80cm
- Akurasi: 3mm
- Sudut pengukuran tertutup: <15 °
- Operasi Saat Ini: <15mA
- Frekuensi Operasi: 40Hz

Robot *wall follower* merupakan sebuah robot yang dirancang untuk mendeteksi keberadaan halangan/benda dalam suatu tempat. Rancangan robot *wall follower* pendeteksi keberadaan halangan terdiri dari perangkat keras dan perangkat lunak yang satu dan lainnya saling berhubungan dan saling mendukung.



Gambar 18. Robot *Wall Follower*

Robot *wall follower* pendeteksi keberadaan halangan menggunakan sensor ultrasonik SR04 untuk mendeteksi halangan yang ada di depannya dengan cara memancarkan sinyal. Sensor ultrasonik SR04 bekerja dengan memancarkan gelombang ultrasonik lalu mendeteksi pantulannya. Selain itu juga menggunakan motor DC sebagai penggerak utama robot serta mikrokontroler Arduino uno yang memiliki kelebihan pada port ADC 8 channel 10-bit sehingga dapat langsung dihubungkan dengan sensor dan pada setiap portnya dapat langsung dihubungkan pada driver motor DC. Mikrokontroller Arduino Uno juga memiliki aplikasi teknologi RISC dengan kecepatan maksimal 16 MHz yang membuatnya lebih cepat dibandingkan dengan varian lainnya.

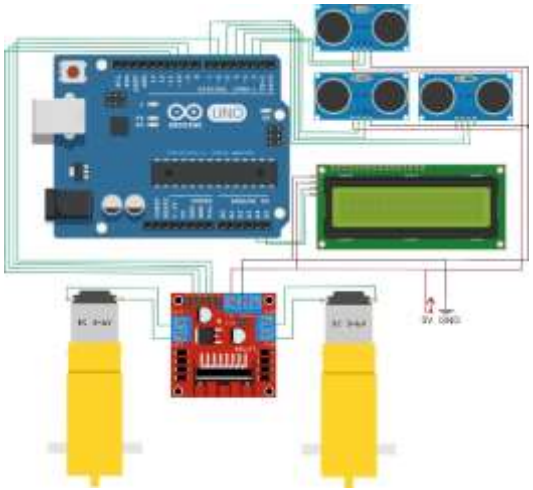
Dalam proses pendeteksian, robot berjalan dengan cara menelusuri dinding/halangan menggunakan sensor ultrasonik pada robot sehingga robot terus berjalan menelusuri halangan tanpa harus menabrak halangan, jadi robot dapat terus berjalan mencari objek. Sensor ultrasonik SR04 mampu mendeteksi adanya objek dikisaran antara 3cm – 3m, jarak yang dideteksi sensor menjadi acuan bagi robot untuk menentukan arah belokan, sehingga robot tidak dapat menabrak halangan dan dapat terus berjalan.

C. Alat dan Bahan

1. Trainer Robotika 1 buah
2. Komputer/laptop 1 buah
3. Kabel USB Arduino 1 buah
4. Jumper Male-male 12 buah

D. Langkah Percobaan

1. Buatlah rangkaian robot *wall follower* dengan mengikuti skema berikut:



Gambar 19. Skema rangkaian robot *wall follower*

Tabel 3 1. Konfigurasi pin rangkaian robot *wall follower*

Arduino Uno	Antarmuka
Pin 1	Triger Ultrasonic Depan
Pin 2	Echo Ultrasonic Depan
Pin 3	Triger Ultrasonic Kiri
Pin 4	Echo Ultrasonic Kiri
Pin 5	Triger Ultrasonic Kanan
Pin 6	PWM Motor IN1
Pin 7	Echo Ultrasonic Kanan
Pin 9	PWM Motor IN2
Pin 10	PWM Motor IN3
Pin 11	PWM Motor IN4
Pin A4	SDA i2C LCD
Pin A5	SCL i2C LCD

Rangkaian tersebut dapat dipraktikkan pada *trainer* dengan memasang *jumper* sebagai berikut:

a. Hubungkan pin 1 Arduino Uno ke Pin Triigger sensor Ultrasonic depan.
b. Hubungkan pin 2 Arduino Uno ke Pin Echo sensor Ultrasonic depan.
c. Hubungkan pin 3 Arduino Uno ke Pin Triigger sensor Ultrasonic Kiri.
d. Hubungkan pin 4 Arduino Uno ke Pin Echo sensor Ultrasonic Kiri.
e. Hubungkan pin 5 Arduino Uno ke Pin Triigger sensor Ultrasonic Kanan.
f. Hubungkan pin 7 Arduino Uno ke Pin Echo sensor Ultrasonic Kanan.
g. Hubungkan pin 6 Arduino Uno ke Pin IN1 driver motor L298N.

h. Hubungkan pin 9 Arduino Uno ke Pin IN2 driver motor L298N.

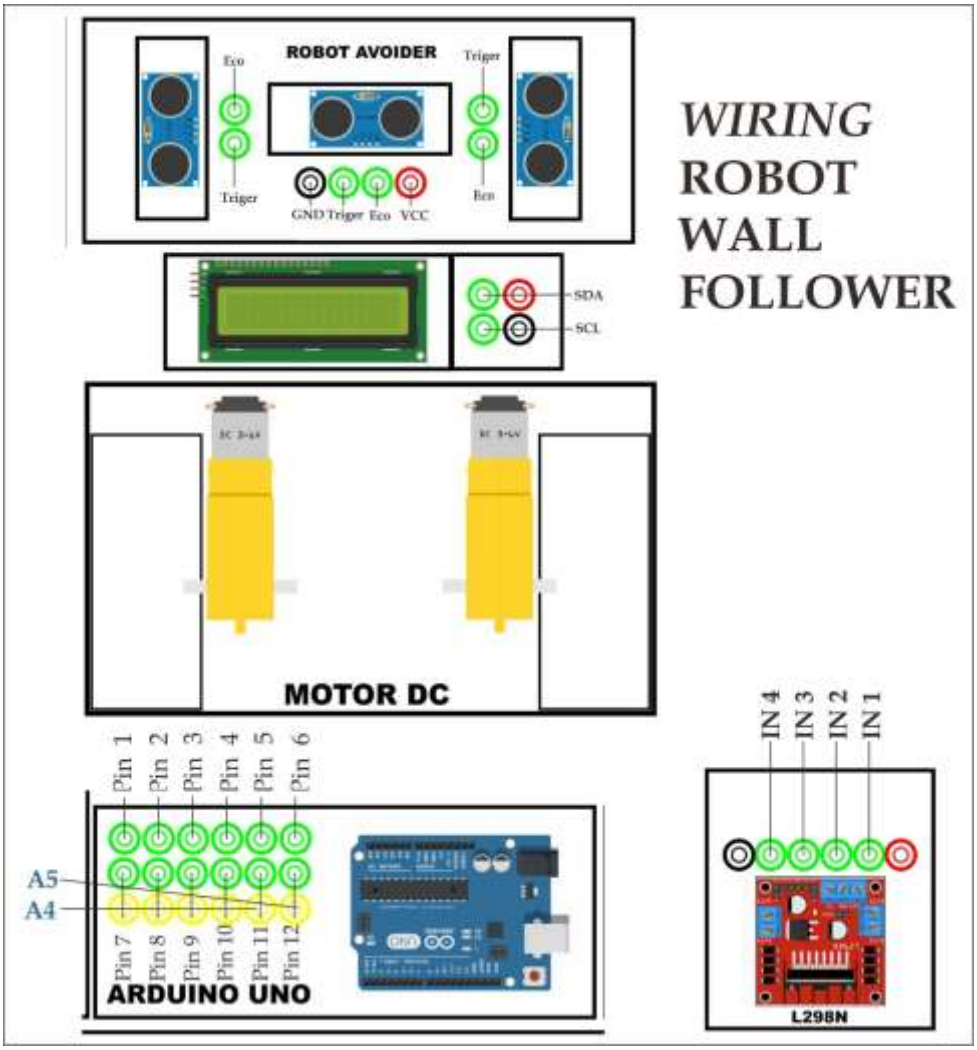
i. Hubungkan pin 10 Arduino Uno ke Pin IN3 driver motor L298N.

j. Hubungkan pin 11 Arduino Uno ke Pin IN4 driver motor L298N.

k. Hubungkan pin A4 Arduino Uno ke Pin SDA LCD 16X2.

l. Hubungkan pin A5 Arduino Uno ke Pin SCL LCD 16X2.

Berikut gambar pemasangan jumper pada trainer



Gambar 20. *Wiring* Robot *Wall Follower* pada trainer

2. Buka aplikasi Arduino IDE di Komputer/Laptop dan masukkan Sketch program berikut:

```
#define trigPin1 1
#define echoPin1 2
#define trigPin2 3
#define echoPin2 4
#define trigPin3 5
#define echoPin3 7
```

```
int mtrkanan1 = 6;
int mtrkanan2 = 9;
int mtrkiri1 = 10;
int mtrkiri2 = 11;

int kecmtr1 = 10;
int kecmtr2 = 10;

long duration1, distance1;
long duration2, distance2;
long duration3, distance3;

void setup() {

  Serial.begin(9600);

  pinMode(trigPin1,OUTPUT);
  pinMode(echoPin1,INPUT);
  pinMode(trigPin2,OUTPUT);
  pinMode(echoPin2,INPUT);
  pinMode(trigPin3,OUTPUT);
  pinMode(echoPin3,INPUT);

  pinMode(mtrkanan1,OUTPUT);
  pinMode(mtrkanan2,OUTPUT);
  pinMode(mtrkiri1,OUTPUT);
  pinMode(mtrkiri2,OUTPUT);
  pinMode(kecmtr1,OUTPUT);
  pinMode(kecmtr2,OUTPUT);

}

void loop() {


  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1,LOW);
  duration1 = pulseIn(echoPin1, HIGH);
  distance1 = (duration1/2) / 29.1;

  digitalWrite(trigPin2,LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin2,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin2,LOW);
  duration2 = pulseIn(echoPin2,HIGH);
  distance2 = (duration2/2) / 29.1;

  digitalWrite(trigPin3,LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin3,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin3,LOW);
  duration3 = pulseIn(echoPin3,HIGH);
  distance3 = (duration3/2) / 29.1;
```

```
Serial.print(distance1);
Serial.print(" ");
Serial.print(distance2);
Serial.print(" ");
Serial.println(distance3);


if ((distance1 > 20)&&(distance2 > 20)&&(distance3 > 20)) {
analogWrite(kecmtr1,250);
digitalWrite(mtrkanan1,HIGH);
digitalWrite(mtrkanan2,LOW);

analogWrite(kecmtr2,250);
digitalWrite(mtrkiri1,LOW);
digitalWrite(mtrkiri2,HIGH);
}

if ((distance1 > 20)&&(distance2 < 20)&&(distance3 > 20)) {
analogWrite(kecmtr1,250);
digitalWrite(mtrkanan1,LOW);
digitalWrite(mtrkanan2,HIGH);

analogWrite(kecmtr2,250);
digitalWrite(mtrkiri1,LOW);
digitalWrite(mtrkiri2,HIGH);
}
if ((distance1 < 20)&&(distance2 < 20)&&(distance3 > 20)) {
analogWrite(kecmtr1,250);
digitalWrite(mtrkanan1,HIGH);
digitalWrite(mtrkanan2,LOW);

analogWrite(kecmtr2,250);
digitalWrite(mtrkiri1,HIGH);
digitalWrite(mtrkiri2,LOW);
}
if ((distance1 > 20)&&(distance2 < 20)&&(distance3 < 20)) {
analogWrite(kecmtr1,250);
digitalWrite(mtrkanan1,LOW);
digitalWrite(mtrkanan2,HIGH);

analogWrite(kecmtr2,250);
digitalWrite(mtrkiri1,LOW);
digitalWrite(mtrkiri2,HIGH);
}
if ((distance1 < 20)&&(distance2 < 20)&&(distance3 < 20)) {
analogWrite(kecmtr1,250);
digitalWrite(mtrkanan1,LOW);
digitalWrite(mtrkanan2,HIGH);

analogWrite(kecmtr2,250);
digitalWrite(mtrkiri1,HIGH);
digitalWrite(mtrkiri2,LOW);
}
if ((distance1 < 20)&&(distance2 > 20)&&(distance3 > 20)) {
analogWrite(kecmtr1,250);
digitalWrite(mtrkanan1,HIGH);
digitalWrite(mtrkanan2,LOW);

analogWrite(kecmtr2,250);
```

```
  digitalWrite(mtrkiri1,HIGH);
  digitalWrite(mtrkiri2,LOW);
  }
  if ((distance1 > 20)&&(distance2 > 20)&&(distance3 < 20)) {
  analogWrite(kecmtr1,250);
  digitalWrite(mtrkanan1,LOW);
  digitalWrite(mtrkanan2,HIGH);

  analogWrite(kecmtr2,250);
  digitalWrite(mtrkiri1,LOW);
  digitalWrite(mtrkiri2,HIGH);
  }
}
```

3. Compile program dengan menekan tombol *Verify/Compile* pada aplikasi Arduino IDE atau dengan menekan tombol Ctrl + R pada *keyboard* untuk memastikan program berhasil dan tidak terdapat kesalahan.
4. Hubungkan Laptop dengan Arduino UNO menggunakan Kabel USB Arduino
5. Upload program pada computer/laptop ke Arduino Uno dengan menekan tombol Upload pada aplikasi Arduino IDE atau dengan menekan tombol Ctrl + U pada *keyboard*.

E. Hasil Percobaan

1. Kondisi awal *Wall Follower Robot* dalam keadaan tidak bergerak ;
2. Penghalang Sensor Ultrasonik bagian depan dilepas, Motor kiri dan kanan dalam kondisi bergerak;
3. Penghalang Sensor Ultrasonik bagian kiri dilepas, Motor kiri Bergerak mundur dan motor kanan dalam kondisi bergerak maju;
4. Penghalang Sensor Ultrasonik bagian kanan dilepas, Motor kanan Bergerak mundur dan motor kiri dalam kondisi bergerak maju;
5. Semua penghalang Sensor Ultrasonik dipasang, Motor kiri dan motor kanan dalam kondisi bergerak mundur;

PERCOBAAN III

ANTARMUKA ROBOT MOBILE *(ROBOT PEMADAM)*

PERCOBAAN III | ANTARMUKA ROBOT MOBILE *(ROBOT PEMADAM)*

A. Tujuan Percobaan

1. Mahasiswa mampu menjelaskan prinsip kerja Flame Sensor/Sensor Api.
2. Mahasiswa mampu mengimplementasikan sensor Api sebagai robot Pemadam pada rangkaian Trainer robotika.
3. Mahasiswa mampu melakukan pemograman sensor apisebagai robot pemadam pada Aplikasi Arduino IDE.

B. Teori Dasar

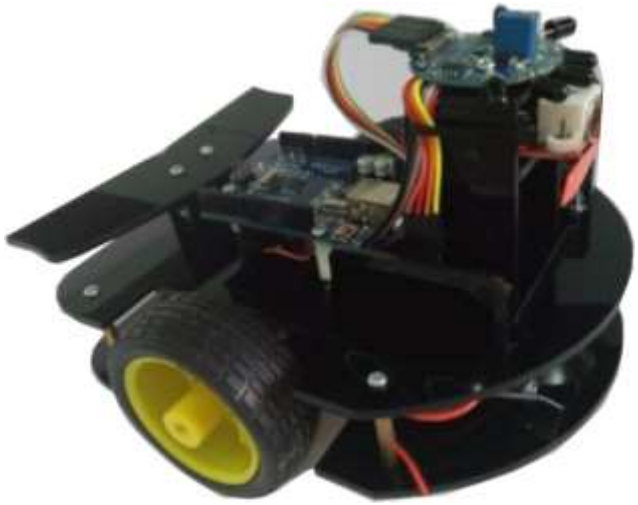
Flame Detector merupakan sensor yang mempunyai fungsi sebagai pendeteksi nyala api yang dimana api tersebut memiliki panjang gelombang antara 760nm – 1100nm. Sensor ini menggunakan infrared sebagai tranduser dalam mensensing kondisi nyala api.

Gambar 21. Sensor Api

Pertandingan kompetisi robot kebanyakan digunakan pendeteksian akan nyala api misalnya lilin masih tetap jadi salah satu aturan yang umum dalam kompetensi lomba yang tidak akan pernah ditinggalkan. Karena itulah sensor ini mempunyai peran yang vital yang berfungsi sebagai "mata" bagi robot dalam menyelesaikan tugasnya menemukan posisi nyala api.

Kompetisi Robot Indonesia atau KRI baik berbentuk laba-laba maupun seperti tank. Selain itu sensor ini sering juga digunakan untuk mendeteksi api pada ruangan di perkantoran, apartemen, maupun di perhotelan. Suhu normal pembacaan normal sensor ini yaitu pada 25 – 85°C dengan besar sudut pembacaan pada 60°. Dengan memperhatikan jarak sensing antara objek yang

akan disensing dengan sensor tidak boleh terlalu dekat, yang berakibat lifetime sensor yang cepat rusak.



Gambar 22. Contoh robot wall follower

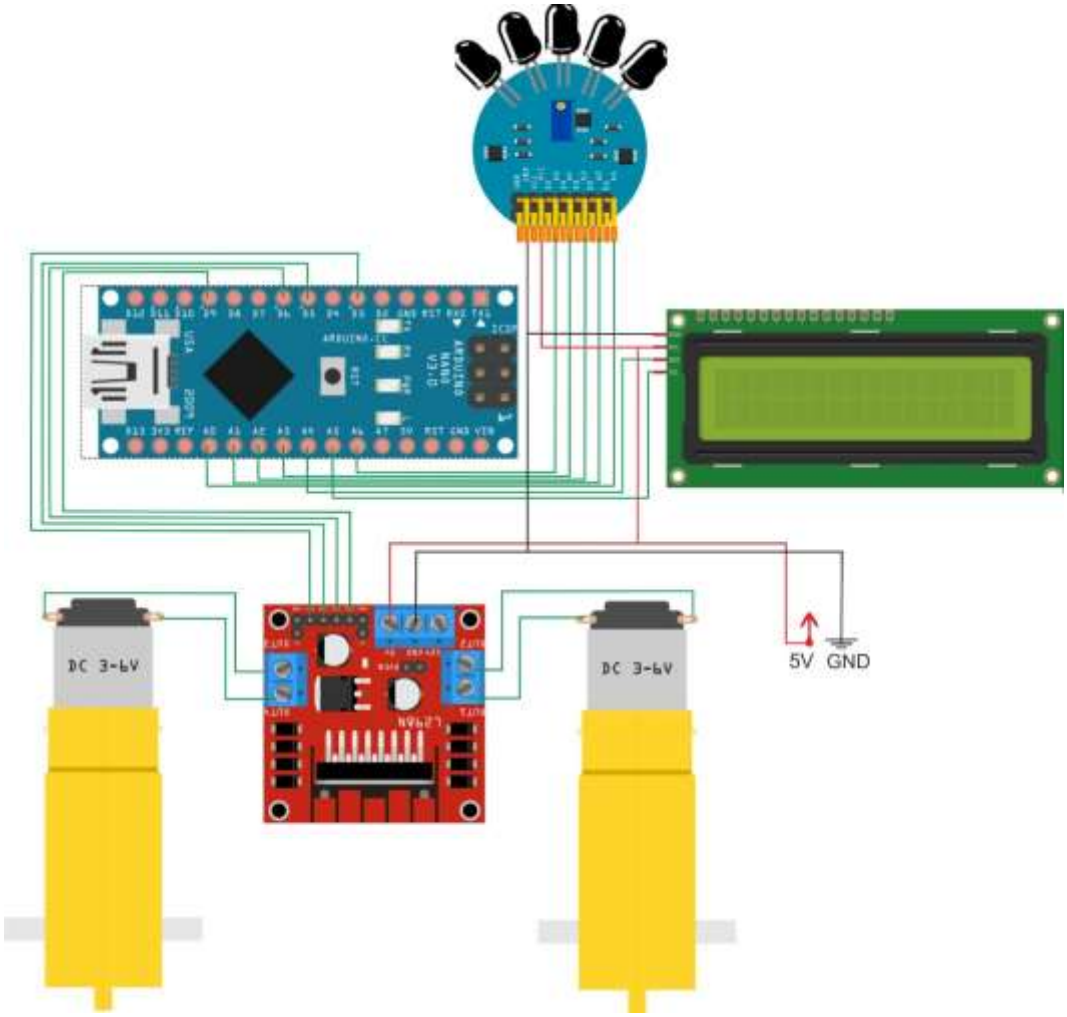
Cara kerja sensor ini yaitu dengan mengidentifikasi atau mendeteksi nyala api dengan menggunakan metode optik. Pada sensor ini menggunakan tranduser yang berupa infrared (IR) sebagai sensing sensor. Tranduser ini digunakan untuk mendeteksi akan penyerapan cahaya pada panjang gelombang tertentu. Yang dimana memungkinkan alat ini untuk membedakan antara spectrum cahaya pada api dengan spectrum cahaya lainnya seperti spectrum cahaya lampu.

C. Alat dan Bahan

1. Trainer Robotika 1 buah
2. Komputer/laptop 1 buah
3. Kabel USB Arduino 1 buah
4. Jumper Male-male 11 buah

D. Langkah Percobaan

1. Buatlah rangkaian robot *wall follower* dengan mengikuti skema berikut:



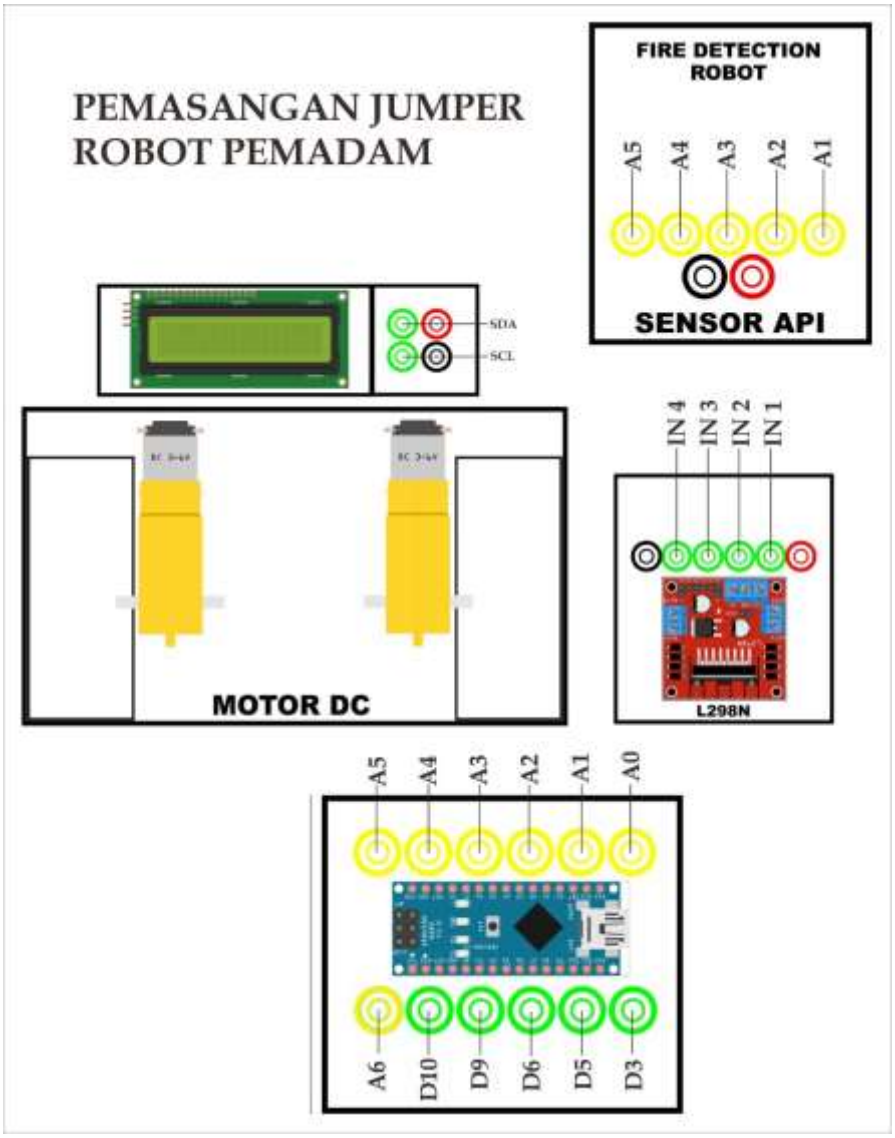
Gambar 23. Skema rangkaian robot pemadam

Konfigurasi pin rangkaian robot pemadam

Arduino Uno	Antarmuka
A0	A1 Flame Sensor
A1	A2 Flame Sensor
A2	A3 Flame Sensor
A3	A4 Flame Sensor
A4	SDA i2C LCD
A5	SCL i2C LCD
A6	A5 Flame Sensor
D3	IN4 Motor DC
D5	IN3 Motor DC
D6	IN2 Motor DC

D9 IN1 Motor DC

Rangkaian tersebut dapat dipraktikkan pada trainer dengan menghubungkan pin A0 Arduino Nano ke Flame sensor A1, pin A1 Arduino Nano ke Flame sensor A2, pin A2 Arduino Nano ke Flame sensor A3, pin A3 Arduino Nano ke Flame sensor A4, pin A4 Arduino Nano ke SDA i2C LCD, pin A5 Arduino Nano ke SCL i2C LCD, pin A6 Arduino Nano ke Flame sensor A5, pin D3 Arduino Nano ke IN4 Driver motor, pin D5 Arduino Nano ke IN3 Driver motor, pin D6 Arduino Nano ke IN2 Driver motor, pin D9 Arduino Nano ke IN1 Driver motor, kemudian Red to 5V , dan terakhir Black to Ground, seperti gambar pemasangan *jumper* pada trainer berikut ini:



Gambar 24. Rangkaian Robot pemadam pada trainer

2. Buka aplikasi Arduino IDE di Komputer/Laptop dan masukkan Sketch program berikut:

Program 1

```
//inisialisai motor servo
#include <Servo.h>
#define RODA1 4
#define RODA2 5
Servo Roda1; Servo Roda2;
//definisi pin kipas dan pin buzzer
#define kipas 6
#define buzzer 3
bool keadaan;
//KALIBRASI KECEPATAN DAN SETPOIN SENSOR
int kecepataan =600; // NILAI GERAK BELOK 90* (500-1000);
int rodakanan  =93;  // NILAI BERHENTI RODA KANAN (+-90);
int rodakiri   =95;  // NILAI BERHENTI RODA KIRI (+-90);
int setpoin    =7;   // NILAI TOLERANSI JARAK HINDAR (7-15);
int lintasan   =200; // NILAI TOLERANSI WARNA LINTASAN (0-1024)
int api        =500; // NILAI TOLERANSI CAHAYA API (0-1024)

//inisialisasi sensor 1 sebelah kiri
#include <NewPing.h>
#define echo1    10
#define trigger1 9
NewPing sonar1(trigger1,echo1,100);

//inisialisasi sensor 2 sebelah depan
#define echo2    12
#define trigger2 11
NewPing sonar2(trigger2,echo2,100);

//inisialisasi sensor 2 sebelah kanan
#define echo3    8
#define trigger3 7
NewPing sonar3(trigger3,echo3,100);


void setup() {
 Serial.begin(9600); // serial monitor "menampilkan data sensor"
 diam(3000);
 Roda1.attach(RODA1);  //pin motor servo
 Roda2.attach(RODA2);
 pinMode(kipas,INPUT);
 pinMode(buzzer,OUTPUT);
 keadaan = 0;
 }

void loop() {
// sensor mengukur dan membaca
int jarak1 =sonar1.ping_cm(); // sensor kiri
int jarak2 =sonar2.ping_cm(); // sensor depan
int jarak3 =sonar3.ping_cm(); // sensor kanan
int base    = analogRead(A0); // pin sensor tcr5000
int api1    = analogRead(A1); // pin sensor Api A1-A5
int api2    = analogRead(A2);
int api3    = analogRead(A3);
```

```
int api4    = analogRead(A4);
int api5    = analogRead(A5);
//sensor menampilkan data yang dibaca
Serial.print(jarak1);
Serial.print("   ||   ");
Serial.print(jarak2);
Serial.print("   ||   ");
Serial.print(jarak3);
Serial.print("   ||   ");
Serial.print(api3);
Serial.print("   ||   ");
Serial.println(base);

//PEMIS 1 jika api terdeteksi maka exsekusi
if ((api2>api||api3>api||api4>api) && (jarak2>1&&jarak2<15)){
digitalWrite(buzzer,HIGH);  //buzzer bunyi
pinMode(kipas, OUTPUT); digitalWrite(kipas,LOW);//kipas bergerak
diam(2000); // panggil diam "lihat inisialisasi progam pangillan
void diam(x);"
tiup(kecepataan);  //panggil tiup
diam(1000); //panggil diam
keadaan = 1;
}

//PREMIS 2 Jika api tidak terdeteksi maka exsekusi
else{
digitalWrite(buzzer,LOW); // buzzer mati
pinMode(kipas, INPUT); digitalWrite(kipas,HIGH); // kipas berhenti
if (base>lintasan && keadaan > 0){
  maju();
  delay(kecepataan);
  diam (100000);//berhenti total
}
//jika sensor kiri terdeteksi maka belok kanan
else if (jarak1<setpoin&&jarak1>1){
  kanan(); //panggil kanan
  delay(50);
  }
//jika sensor kanan terdeteksi maka belok kiri
else if (jarak3<setpoin&&jarak3>1){
  kiri();
  delay(50);
  }
//jika sensor depan terdeteksi maka balik (balik belum ditentukan)
else if (jarak2<setpoin&&jarak2>1){
  //jika sensor kiri < sensor kanan maka ditentukan "balik kanan"
  if (jarak1<jarak3){
  balik_kanan();
  delay(kecepataan);
  }
  //jika sensor kanan > sensor kanan maka ditentukan "balik kiri"
  else if (jarak1>jarak3){
  balik_kiri();
  delay(kecepataan);
}
}
//JIKA NIALI SET POIN SEMUA SENSOR TIDAK TERLAMPAUI MAKA "ROBOT
MAJU"
else{
```

```
  maju();
  delay(100);
}
}
}

// inisialisasi progam pangillan
void maju(){          //SEMUA RODA MAJU
  Roda1.write(380);
  Roda2.write(-380);
}
void mundur(){        //SEMUA RODA MUNDUR
  Roda1.write(-380);
  Roda2.write(380);
}
void balik_kiri(){
  Roda1.write(-380);   //RODA1 MUNDUR
  Roda2.write(-380);   //RODA2 MAJU
}
void balik_kanan(){
  Roda1.write(380);    //RODA1 MAJU
  Roda2.write(380);    //RODA2 MUNDUR
}
void kanan(){
 Roda1.write(380);       //RODA1 MAJU
 Roda2.write(rodakanan); //RODA2 BERHENTI
}
void kiri(){
 Roda1.write(rodakiri); //RODA1 BERHENTI
 Roda2.write(-380);     //RODA2 MAJU
}
void diam(int x){
  Roda1.write(93);  //RODA KANAN BERHENTI
  Roda2.write(95);  //RODA KIRI BERHENTI
  delay(x);         //SEMUA RODA BERHENTI
  }
void tiup(int x){   // ROBOT BERHENTI DAN MENIUP KEKANAN DAN
KRKIRI
balik_kanan();
delay(x/4);
diam(x);
balik_kiri();
delay(x/2);
diam(x);
balik_kanan();
delay(x/2);
diam(x);
balik_kiri();
delay(x/4);
diam(x);
}
```

Program 2

```
#define KIPAS   6
#define BUZZER  3

void setup() {
  pinMode(KIPAS, OUTPUT);
  pinMode(BUZZER, OUTPUT);
```

```
}

void loop() {
  digitalWrite(KIPAS, HIGH); //KIPAS BERGERAK
  digitalWrite(BUZZER,HIGH); //BUZZER BERBUNYI
  delay(1000);
  digitalWrite(KIPAS, LOW); //KIPAS BERHENTI
  digitalWrite(BUZZER,LOW); //BUZZER DIAM
  delay(3000);
  // JIKA KIPAS DAN BUZZER BEKERJA TERBALIK MAKA DI PROGAM DIRUBAH
  //MISAL KIPAS BEKERJA KETIKA DI LOGIKA "LOW"
}
```

Program 3

```
//TES MENCARI TITIK BERHENTI RODA1 & RODA2//


#include <Servo.h>
#define RODA1 4
#define RODA2 5
Servo Roda1; Servo Roda2;

void setup(){
 Roda1.attach(RODA1);
 Roda2.attach(RODA2);

}
void loop(){
  Roda1.write(90);  // rubah angka 110-70 samapai motor berhenti
  Roda2.write(90);  // misal 80,85,90,95,100

  }
```

3. Compile program dengan menekan tombol Verify/Compile pada aplikasi Arduino IDE atau dengan menekan tombol Ctrl + R pada keyboard untuk memastikan program berhasil dan tidak terdapat kesalahan.
4. Hubungkan Laptop dengan Arduino Nano menggunakan Kabel USB Arduino
5. Upload program pada computer/laptop ke Arduino Nano dengan menekan tombol Upload pada aplikasi Arduino IDE atau dengan menekan tombol Ctrl + U pada keyboard.

E. Hasil Percobaan

1. Kondisi awal Robot Pemadam dalam keadaan tidak bergerak ;
2. Api diletakkan pada posisi lurus/depan sensor api, kondisi motor kiri dan motor kanan dalam kondisi bergerak ;
3. Api diletakkan pada posisi kiri sensor api, kondisi motor kiri bergerak mundur dan kondisi motor kanan dalam kondisi bergerak maju ;

4. Api diletakkan pada posisi kanan sensor api, kondisi motor kanan bergerak mundur dan kondisi motor kiri dalam kondisi bergerak maju ;

Catatan* : Perangkat tambahan seperti LED, Buzzer dan Servo di Uji secara terpisah

PERCOBAAN IV

ANTARMUKA ROBOT MOBILE *(LINE FOLLOWER)*

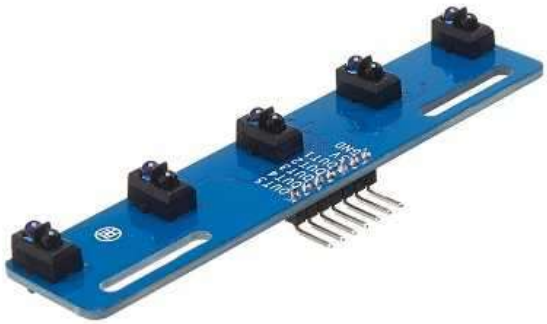
PERCOBAAN IV: ANTARMUKA ROBOT MOBILE *(LINE FOLLOWER)*

A. Tujuan Percobaan

1) Mahasiswa mampu menjelaskan prinsip kerja Sensor Photodioda.
2) Mahasiswa mampu mengimplementasikan sensor photodioda sebagai robot *line follower* pada rangkaian Trainer robotika.
3) Mahasiswa mampu melakukan pemograman sensor photodioda sebagai robot *line follower* pada Aplikasi Arduino IDE.

B. Teori Dasar

Photodiode atau dalam bahasa Indonesia sering disebut dengan Photodioda adalah komponen Elektronika yang dapat mengubah cahaya menjadi arus listrik. Photodioda merupakan komponen aktif yang terbuat dari bahan semikonduktor dan tergolong dalam keluarga Dioda. Seperti Dioda pada umumnya, Photodiode atau Photodioda ini memiliki dua kaki terminal yaitu kaki terminal Katoda dan kaki terminal Anoda, namun Photodioda memiliki Lensa dan Filter Optik yang terpasang dipermukaannya sebagai pendeteksi cahaya.

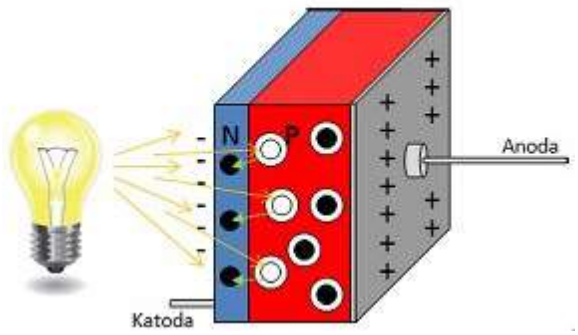


Gambar 25. Modul sensor TCR

Cahaya yang dapat dideteksi oleh Photodioda diantaranya seperti Cahaya Matahari, Cahaya Tampak, Sinar Inframerah, Sinar Ultra-violet hingga

sinar X. Oleh karena itu, Photodiode atau Photodioda yang dapat mendeteksi berbagai Cahaya ini telah banyak diaplikasikan ke berbagai perangkat Elektronika dan listrik seperti Penghitung Kendaraan, Sensor Cahaya Kamera, Alat-alat medis, Scanner Barcode dan peralatan keamanan.

Photodiode terdiri dari satu lapisan tipis semikonduktor tipe-N yang memiliki kebanyakan elektron dan satu lapisan tebal semikonduktor tipe-P yang memiliki kebanyakan Hole. Lapisan semikonduktor tipe-N adalah Katoda sedangkan lapisan semikonduktor tipe-P adalah Anoda.

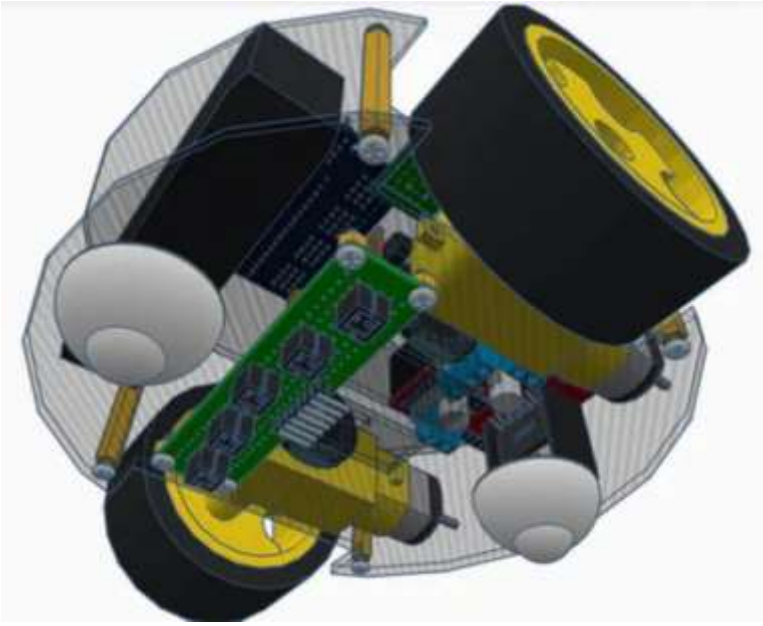


Gambar 26. Prinsip kerja sensor

Photodiode terkena cahaya, Foton yang merupakan partikel terkecil cahaya akan menembus lapisan semikonduktor tipe-N dan memasuki lapisan semikonduktor tipe-P. Foton-foton tersebut kemudian akan bertabrakan dengan elektron-elektron yang terikat sehingga elektron tersebut terpisah dari intinya dan menyebabkan terjadinya *Hole*. Elektron terpisah akibat tabrakan dan berada dekat persimpangan PN (PN junction) akan menyeberangi persimpangan tersebut ke wilayah semikonduktor tipe-N. Hasilnya, Elektron akan bertambah di sisi semikonduktor N sedangkan sisi semikonduktor P akan kelebihan *Hole*. Pemisahan muatan positif dan negatif ini menyebabkan perbedaan potensial pada persimpangan PN. Ketika kita hubungkan sebuah beban ataupun kabel ke Katoda (sisi semikonduktor N) dan Anoda (sisi semikonduktor P), Elektron akan mengalir melalui beban atau kabel tersebut dari Katoda ke Anoda atau biasanya kita sebut sebagai aliran arus listrik.

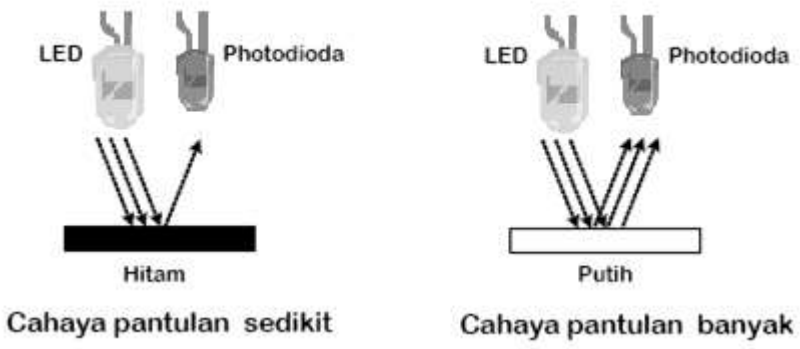
Line follower robot adalah sebuah robot yang dapat mengikuti garis secara otomatis. Robot didukung oleh rangkain komponen elektronika yang dilengkapi dengan roda dan digerakan oleh motor. Pengendalian kecepatan

sangat bergantung pada batas putaran dan pergesekan antara ban robot dengan lantainya. Robot tersebut dirancang untuk bernavigasi dan bergerak secara otomatis mengikuti sebuah alur garis yang dibuat. Untuk membaca garis, robot dilengkapi dengan sensor yang diletakkkan diujung depan dari robot tersebut. Line follower robot ini memiliki jenis dan bentuk serta memiliki beberapa sistem penggerak dan pengendali sebagai pengatur kinerja yang beraneka ragam sesuai dengan kreatifitas pembuatnya.



Gambar 27. Robot line follower 5 sensor

Line Follower Robot merupakan robot yang dapat mengikuti garis, sebagai peng-indera robot ini menggunakan sensor yang berguna untuk membaca atau mendeteksi garis, warna garis yang biasa di gunakan adalah putih untuk base/dasar lapangan warna hitam dan garis hitam untuk base/dasar lapangan putih, dengan dua warna berbeda itu sensor dapat mendeteksi garis dengan cara membedakan warna yang gelap dan warna yang terang. Untuk sensor yang biasa digunakan untuk membaca garis diantaranya sensor photodioda, phototransistor, infrared maupun sensor LDR (Light Dependent Resistor), untuk robot yang saya buat kali ini menggunakan sensor photodioda kenapa sensor itu, karena sensor ini mudah didapat yang banyak dijual di toko komponen elektronika dan harga yang juga terjangkau tentunya.



Gambar 28. Ilustrasi mekanisme pembacaan garis

Kemudian untuk pemeroses atau otak dari robot ini menggunakan mikrokontroler, untuk robot line follower kali ini saya menggunakan mikrokontroler arduino yaitu arduino uno fungsi dari mikro untuk memproses data dari sensor dan kemudian mengontrol motor sebagai aktuator penggerak robot, mikrokontroler ini di isi dengan program atau coding yang akan mengatur robot untuk bernavigasi mengikuti garis.

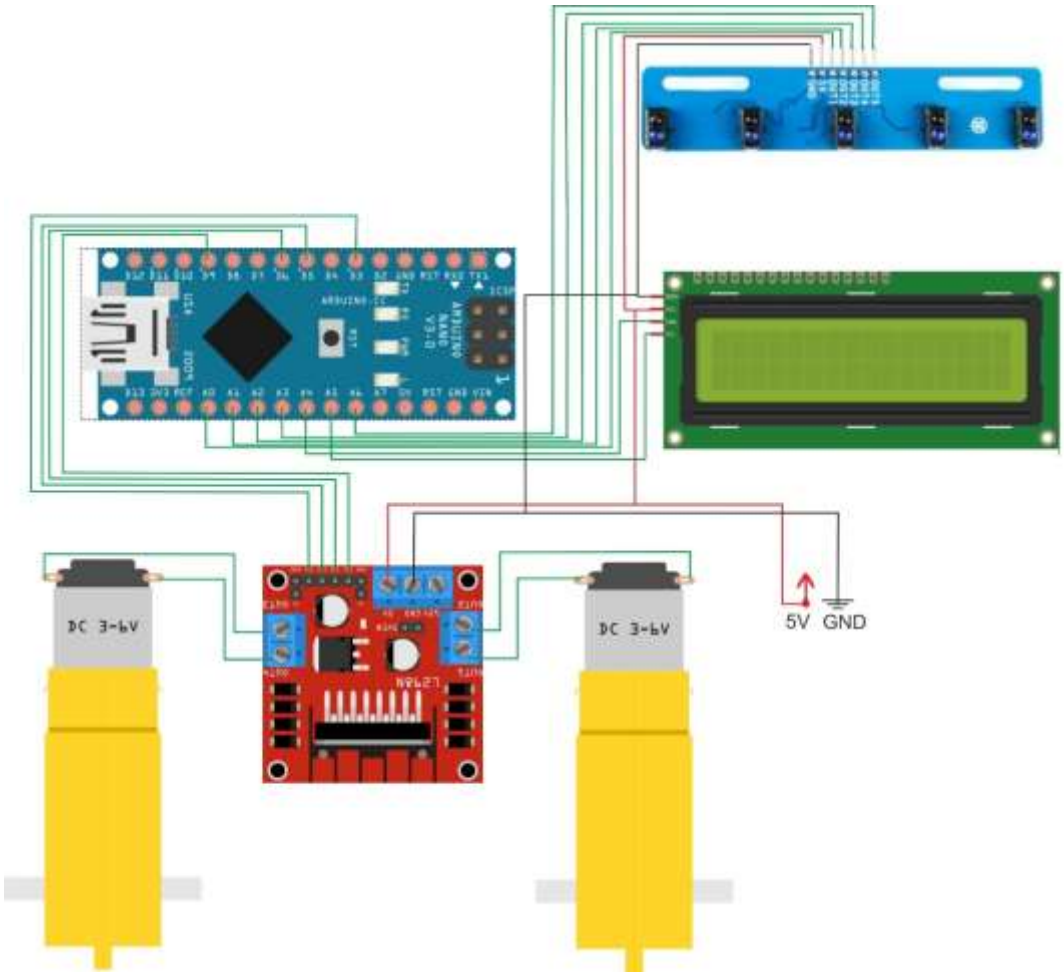
Kondisi ideal robot adalah pada saat sensor berada pada posisi 00100 (nilai 1 di asumsikan sensor mengenai garis) atau NilaiPosisi = 0 dan Setpoint = 0 merupakan posisi ideal robot line follower jika NilaiPosisi tidak = 0 berarti robot tidak pada posisi ideal dan artinya ada nilai error, pada kondisi error output akan mengeluarkan hasil kendalinya sesuai dengan besaran error yang terjadi dan kemudian akan dikalkulasikan dengan nilai base yang ditentukan untuk menjadi PWM ke motor kiri dan kanan.

C. Alat dan Bahan

1) Trainer Robotika 1 buah
2) Komputer/laptop 1 buah
3) Kabel USB Arduino 1 buah
4) Jumper Male-male 11 buah
5) Garis hitam 1 buah

D. Langkah Percobaan

1. Buatlah rangkaian robot *wall follower* dengan mengikuti skema berikut:



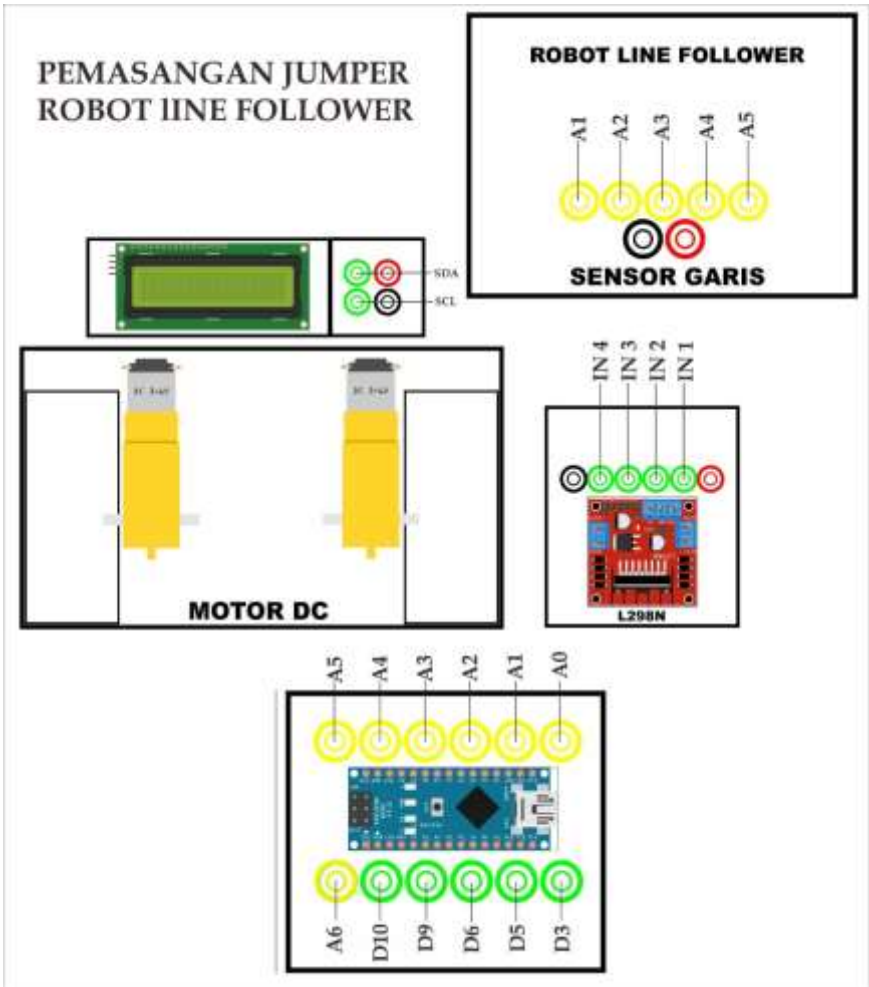
Gambar 29. Skema rangkaian robot line follower

Konfigurasi pin rangkaian robot line follower

Arduino Uno	Antarmuka
A0	Out1 Tracking Sensor
A1	Out2 Tracking Sensor
A2	Out3 Tracking Sensor
A3	Out4 Tracking Sensor
A4	SDA i2C LCD
A5	SCL i2C LCD
A6	Out5 Tracking Sensor
D3	IN4 Motor DC
D5	IN3 Motor DC

D6	IN2 Motor DC
D9	IN1 Motor DC

Rangkaian tersebut dapat dipraktikkan pada trainer dengan menghubungkan pin A0 Arduino Nano ke Out1 Tracking Sensor, pin A1 Arduino Nano ke Out2 Tracking Sensor, pin A2 Arduino Nano ke Out3 Tracking Sensor, pin A3 Arduino Nano ke Out4 Tracking Sensor, pin A4 Arduino Nano ke SDA i2C LCD, pin A5 Arduino Nano ke SCL i2C LCD, pin A6 Arduino Nano ke Out5 Tracking Sensor, pin D3 Arduino Nano ke IN4 Driver motor, pin D5 Arduino Nano ke IN3 Driver motor, pin D6 Arduino Nano ke IN2 Driver motor, pin D9 Arduino Nano ke IN1 Driver motor, kemudian Red to 5V , dan terakhir Black to Ground, seperti gambar pemasangan *jumper* pada trainer berikut ini:



Gambar 30. Rangkaian Robot pemadam pada trainer

2. Buka aplikasi Arduino IDE di Komputer/Laptop dan masukkan Sketch program berikut:

```
void setup()
{
// put your setup code here, to run once:
// SENSOR
  pinMode(A0, INPUT_PULLUP);
  pinMode(A1, INPUT_PULLUP);
  pinMode(A2, INPUT_PULLUP);
  pinMode(A3, INPUT_PULLUP);
  pinMode(A6, INPUT_PULLUP);
// MOTOR
  pinMode(D3, OUTPUT);
  pinMode(D5, OUTPUT);
  pinMode(D6, OUTPUT);
  pinMode(D7, OUTPUT);
}

void loop() {
  int L= digitalRead(2); //sensor 1
  int AL= digitalRead(3); //sensor 2
  int M= digitalRead(4); //sensor 3
  int AR= digitalRead(5); //sensor 4
  int R= digitalRead(6); //sensor 5
  if (L==HIGH && AL==HIGH && M==LOW && AR==HIGH && R==HIGH)

{
  digitalWrite(7, HIGH);   //motor 2
  digitalWrite(8, LOW);              //FORWARD
  digitalWrite(9, HIGH);   //motor 2
  digitalWrite(10, LOW);
}
  else if(L==HIGH && AL==HIGH && M==HIGH && AR==LOW && R==HIGH)
{
  digitalWrite(7, LOW);   //motor 2
  digitalWrite(8, HIGH);             //RIGHT
  digitalWrite(9, HIGH);   //motor 2
  digitalWrite(10, LOW);
}
  else if(L==HIGH && AL==HIGH && M==LOW && AR==LOW && R==HIGH)
{
  digitalWrite(7, LOW);   //motor 2
  digitalWrite(8, HIGH);             //RIGHT
  digitalWrite(9, HIGH);   //motor 2
  digitalWrite(10, LOW);
}
  else if(L==HIGH && AL==HIGH && M==HIGH && AR==LOW && R==LOW)
{
  digitalWrite(7, LOW);   //motor 2
  digitalWrite(8, HIGH);             //RIGHT
  digitalWrite(9, HIGH);   //motor 2
  digitalWrite(10, LOW);
}
else if(L==HIGH && AL==LOW && M==HIGH && AR==HIGH && R==HIGH)
{
  digitalWrite(7, HIGH);   //motor 2
  digitalWrite(8, LOW);              //LEFT
```

```
  digitalWrite(9, LOW);   //motor 2
  digitalWrite(10, HIGH);
}
  else if(L==HIGH && AL==LOW && M==LOW && AR==HIGH && R==HIGH)
{
  digitalWrite(7, HIGH);   //motor 2
  digitalWrite(8, LOW);              //LEFT
  digitalWrite(9, LOW);   //motor 2
  digitalWrite(10, HIGH);
}
  else if(L==LOW && AL==LOW && M==HIGH && AR==HIGH && R==HIGH)
{
  digitalWrite(7, LOW);   //motor 2
  digitalWrite(8, HIGH);             //RIGHT
  digitalWrite(9, HIGH);   //motor 2
  digitalWrite(10, LOW);
}
  else if(L==LOW && AL==LOW && M==HIGH && AR==LOW && R==LOW)
{
  digitalWrite(7, LOW);   //motor 2
  digitalWrite(8, LOW);              //RIGHT
  digitalWrite(9, LOW);   //motor 2
  digitalWrite(10, LOW);
}
  else if(L==HIGH && AL==HIGH && M==HIGH && AR==HIGH &&
R==HIGH)
{
  digitalWrite(7, LOW);   //motor 2
  digitalWrite(8, HIGH);             //RIGHT
  digitalWrite(9, HIGH);   //motor 2
  digitalWrite(10, LOW);
}
  else if(L==LOW && AL==LOW && M==LOW && AR==LOW && R==LOW)
{
  digitalWrite(7, LOW);   //motor 2
  digitalWrite(8, LOW);              //RIGHT
  digitalWrite(9, LOW);   //motor 2
  digitalWrite(10, LOW);
}

  else
{
  digitalWrite(7, HIGH);   //motor 1
  digitalWrite(8, LOW);                  //STOP
  digitalWrite(9, HIGH  );   //motor 2
  digitalWrite(10, LOW);
}
}
```

3. Compile program dengan menekan tombol Verify/Compile pada aplikasi Arduino IDE atau dengan menekan tombol Ctrl + R pada keyboard untuk memastikan program berhasil dan tidak terdapat kesalahan.
4. Hubungkan Laptop dengan Arduino Nano menggunakan Kabel USB Arduino

5. Upload program pada computer/laptop ke Arduino Nano dengan menekan tombol Upload pada aplikasi Arduino IDE atau dengan menekan tombol Ctrl + U pada keyboard.

E. Hasil Percobaan

1. Kondisi awal Robot Line Follower dalam keadaan tidak bergerak ;
2. Garis hitam diletakkan pada posisi lurus/tengah sensor tracer, kondisi motor kiri dan motor kanan dalam kondisi bergerak maju ;
3. Garis hitam diletakkan pada posisi kiri sensor tracer, kondisi motor kiri bergerak mundur dan kondisi motor kanan dalam kondisi bergerak maju ;
4. Garis hitam diletakkan pada posisi kanan sensor tracer, kondisi motor kanan bergerak mundur dan kondisi motor kiri dalam kondisi bergerak maju ;

PERCOBAAN V

ANTARMUKA ROBOT MOBILE (KONTROL MANUAL JOYSTICK)

PERCOBAAN V| ANTARMUKA ROBOT MOBILE (KONTROL MANUAL JOYSTICK)

A. Tujuan Percobaan

1) Mahasiswa mampu menjelaskan prinsip kerja Joystick.
2) Mahasiswa mampu mengimplementasikan Joystick sebagai robot *mobile* kendali manual pada rangkaian Trainer robotika.
3) Mahasiswa mampu melakukan pemograman joystick sebagai robot manual kendali joystick pada Aplikasi Arduino IDE.

B. Teori Dasar

Modul joystick adalah komponen yang berbentuk seperti tuas atau tongkat yang dapat digerakan ke berbagai arah untuk mendapatkan posisi yang diinginkan. Pada umumnya modul ini memiliki 2 axis yaitu axis X dan axis Y dan 1 push button. Pengaplikasian modul ini banyak dijumpai pada joystick game PlayStasiun, X-Box, pengendali servo motor, kursi motor, dan lain – lain. Modul ini yang banyak dipakai yaitu tipe bi-axial. Tipe joystick ini merupakan tipe yang sama dengan yang digunakan pada gagang kendali analog pada konsol Sony Playstation, X-box.

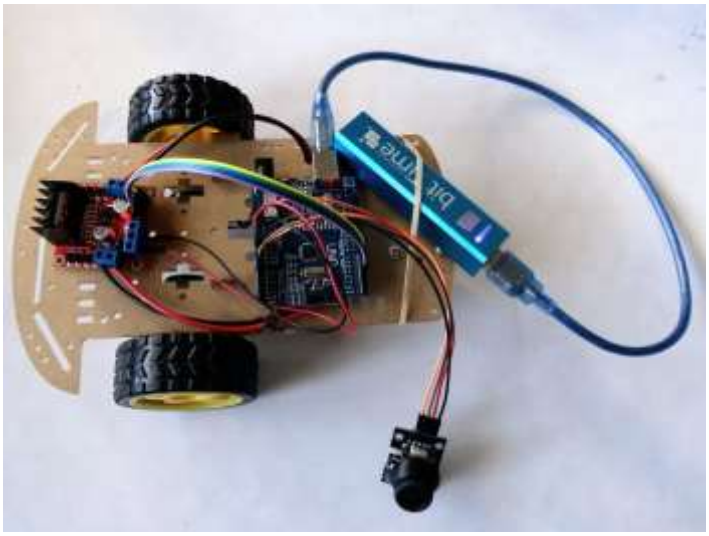
Gambar 31. Modul Joystick dual axis

Spesifikasi dari modul bi-axial

- Terdapat dual-axis X,Y
- Dimensi : 4cm x 2.6cm x 3.2cm
- Terdapat 2 potentiometers untuk 2 axis
- Terdapat 1 switch (push button)

- pin pin kaki: +5Vcc – GND – VRx – VRy – SW

Joystik adalah alat inputan yang berwujud tuas dan dapat bergerak ke segala arah. Joystick pada umumnya digunakan sebagai pelengkap untuk memainkan permainan video yang dilengkapi tombol.



Gambar 32. Robot manual Joystick

Joystik merupakan piranti pengendali tak langsung, gerakan robot dikendalikan oleh gerakan tuas pada joystick absolut atau dengan tekanan Pada tuas. Pada joystick biasanya terdapat tombol yang dapat dipilih atau diaplikasikan dengan papan ketik. Joystick digunakan untuk mengendalikan robot manual pengangkat dan pemindah barang agar bergerak dan dapat memindahkan barang, dalam pengoperasiannya, joystick tidak memerlukan tempat yang luas. Setelah tombol pada joystick ditekan maka data akan dikirimkan menuju mikrokontroler untuk diproses. tombol joystick disambungkan dengan port-port input pada mikrokontroler dan tiap port-port output yang telah disambungkan dengan beban, memberikan perintah dari input joystick setelah diproses oleh mikrokontroler agar beban yang berupa motor dc dapat bergerak.

Joystik menggunakan saklar Push Button di setiap tombolnya, Push Button disini mempunyai dua masukan yakni untuk pemberi input dan kommon (pada perancangan kommon dihubungkan ke ground). Dengan disetnya kommon dengan ground, apabila menekan tombol otomatis ketiga

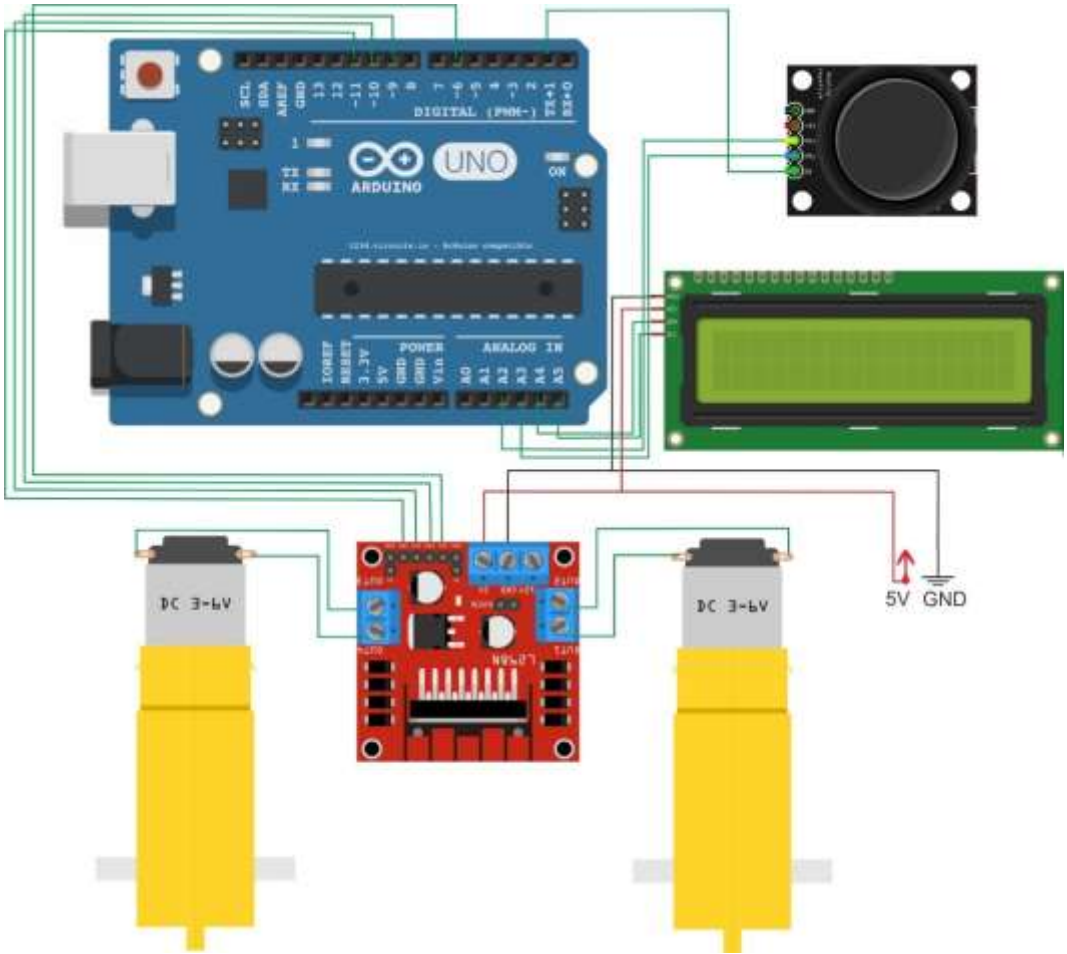
masukan terhubung, dengan kata lain kolom dan baris berlogika '0' perubahan logika inilah yang diproses oleh mikrokontroler.

C. Alat dan Bahan

1) Trainer Robotika 1 buah
2) Komputer/laptop 1 buah
3) Kabel USB Arduino 1 buah
4) Jumper Male-male 9 buah

D. Langkah Percobaan

1. Buatlah rangkaian robot mobile kontrol manual dengan mengikuti skema berikut:

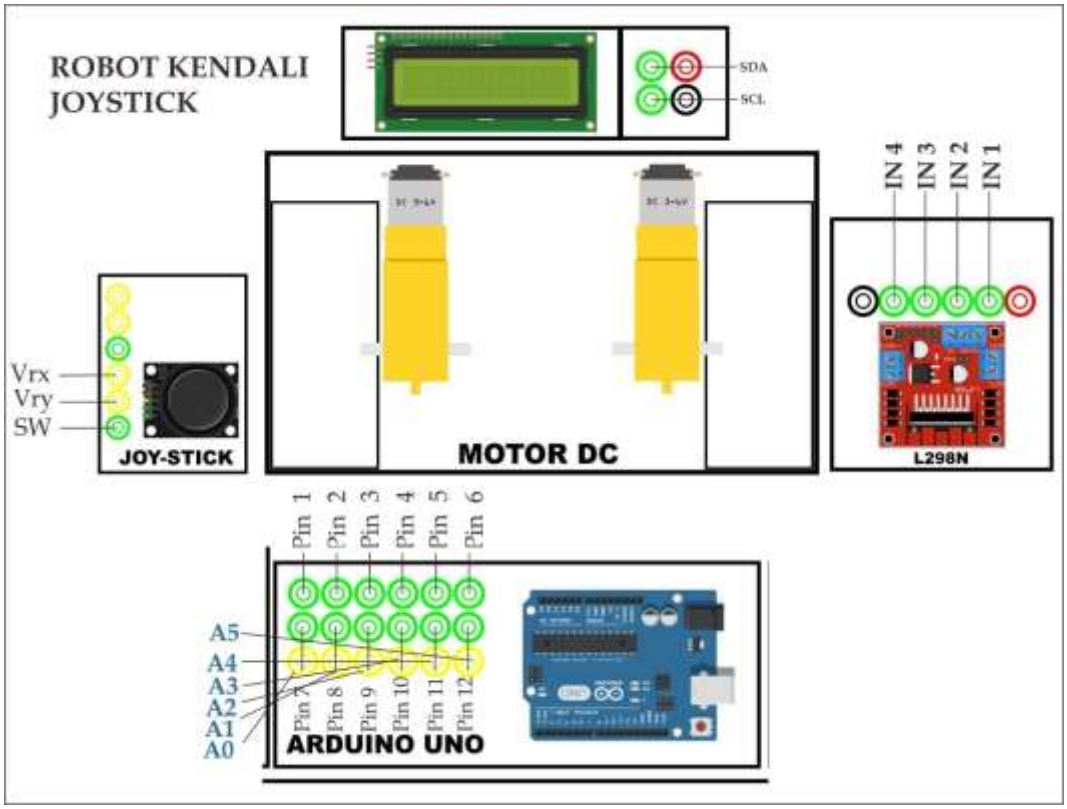


Gambar 33. Skema rangkaian robot manual kendali joystick

Konfigurasi pin rangkaian robot manual kendali joystick

Arduino Uno	**Antarmuka**
A2	VRx joystick
A3	VRy joystick
A4	SDA i2C LCD
A5	SCL i2C LCD
D1	SW Joystick
D6	IN4 Motor DC
D9	IN3 Motor DC
D10	IN2 Motor DC
D11	IN1 Motor DC

Rangkaian tersebut dapat dipraktikkan pada trainer dengan menghubungkan pin A2 Arduino Uno ke VRx Joystick, pin A3 Arduino Uno ke VRy Joystick, pin A4 Arduino Uno ke SDA i2C LCD, pin A5 Arduino Uno ke SCL i2C LCD, pin D1 Arduino Uno ke SW Joystick, pin D6 Arduino Nano ke IN4 Driver motor, pin D9 Arduino Nano ke IN3 Driver motor, pin D10 Arduino Nano ke IN2 Driver motor, pin D11 Arduino Nano ke IN1 Driver motor, kemudian Red to 5V , dan terakhir Black to Ground, seperti gambar pemasangan *jumper* pada trainer berikut ini:



Gambar 34. Rangkaian Robot manual kendali joystick

2. Buka aplikasi Arduino IDE di Komputer/Laptop dan masukkan Sketch program berikut:

```
int joystrick1 = A2;
int joystrick2 = A3;
int val1;
int val2;

void setup() {
  // put your setup code here, to run once:
Serial.begin(9600);
pinMode(6, OUTPUT);
pinMode(9, OUTPUT);
pinMode(10, OUTPUT);
pinMode(11, OUTPUT);
pinMode(A2, INPUT);
pinMode(A3, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  int val1 = analogRead(joystrick1);
  int val2 = analogRead(joystrick2);

 if(val1 > 1 && val1 < 490){
val1 = map(val1, 1, 490, 255, 0);

analogWrite(6,val1);
analogWrite(5,0);
```

```
}

if(val1 > 530 && val1 < 1023){
val1 = map(val1, 530, 1023, 0, 255);

analogWrite(5,val1);
analogWrite(6,0);
}

if(val1 > 491 && val1 < 529){
val1 = map(val1, 491, 529, 0, 0);

analogWrite(5,val1);
analogWrite(6,val1);
}

if(val2 > 1 && val2 < 490){
val2 = map(val2, 1, 490, 255, 0);

analogWrite(10,val2);
analogWrite(9,0);
}

if(val2 > 530 && val2 < 1023){
val2 = map(val2, 530, 1023, 0, 255);

analogWrite(9,val2);
analogWrite(10,0);
}

if(val2 > 491 && val2 < 529){
val2 = map(val2, 491, 529, 0, 0);

analogWrite(9,val2);
analogWrite(10,val2);
}
Serial.println(val1);
Serial.println(val2);
}
```

3. Compile program dengan menekan tombol Verify/Compile pada aplikasi Arduino IDE atau dengan menekan tombol Ctrl + R pada keyboard untuk memastikan program berhasil dan tidak terdapat kesalahan.
4. Hubungkan Laptop dengan Arduino Uno menggunakan Kabel USB Arduino
5. Upload program pada computer/laptop ke Arduino Uno dengan menekan tombol Upload pada aplikasi Arduino IDE atau dengan menekan tombol Ctrl + U pada keyboard.

E. Hasil Percobaan

1. Kondisi awal *mobile robot* (Kontrol Joystick) dalam keadaan tidak bergerak ;
2. Maju pada Joystick Motor Kiri dan Motor kanan bergerak maju;
3. Kiri pada Joystick Motor Kiri berhenti/stop dan Motor kanan bergerak maju;
4. Kanan pada Joystick Motor kanan berhenti/stop dan Motor Kiri bergerak maju;
5. Mundur pada Joystick Motor Kiri dan Motor kanan bergerak mundur;

PERCOBAAN VI

ANTARMUKA ROBOT MOBILE (KONTROL MANUAL SMARTPHONE BLUETOOTH)

PERCOBAAN VI | ANTARMUKA ROBOT MOBILE (KONTROL MANUAL SMARTPHONE BLUETOOTH)

A. Tujuan Percobaan

1) Mahasiswa mampu menjelaskan prinsip kerja bluetooth HC-05.
2) Mahasiswa mampu mengimplementasikan bluetooth HC-05 sebagai robot *mobile* kendali manual pada rangkaian Trainer robotika.
3) Mahasiswa mampu melakukan pemograman bluetooth HC-05 sebagai robot manual kendali joystick pada Aplikasi Arduino IDE.

B. Teori Dasar

Bluetooth adalah protokol komunikasi wireless yang bekerja pada frekuensi radio 2.4 GHz untuk pertukaran data pada perangkat bergerak seperti PDA, laptop, HP, dan lain-lain1 . Salah satu hasil contoh modul Bluetooth yang paling banyak digunakan adalah tipe HC-05. modul Bluetooth HC-05 merupakan salah satu modul Bluetooth yang dapat ditemukan dipasaran dengan harga yang relatif murah. Modul Bluetooth HC-05 terdiri dari 6 pin konektor, yang setiap pin konektor memiliki fungsi yang berbeda - beda.

Gambar 35. Bluetooth HC-05

Bluetooth adalah protokol komunikasi wireless yang bekerja pada frekuensi radio 2.4 GHz untuk pertukaran data pada perangkat bergerak

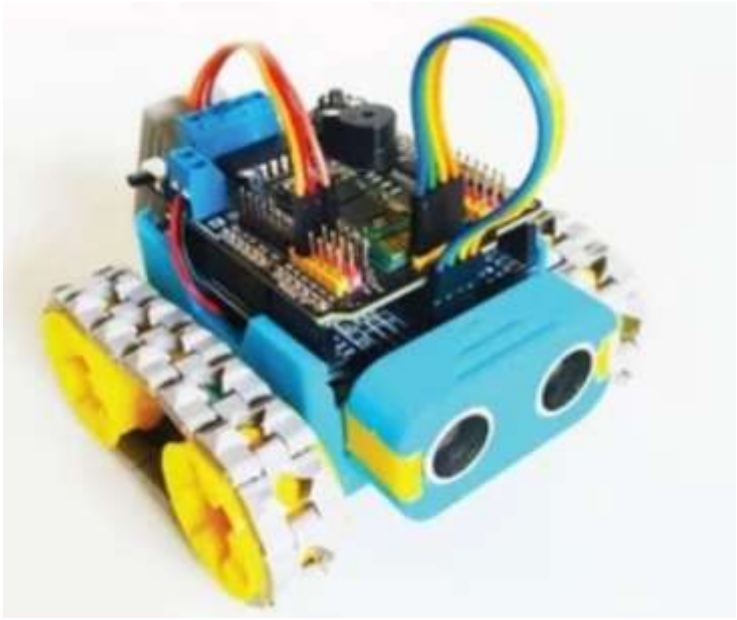
seperti pada,laptop, HP, dan lain-lain. Salah satu hasil contoh modul Bluetooth yang paling banyak digunakan adalah tipe HC-05. Modul Bluetooth HC-05 merupakan modul Bluetooth yang bisa menjadi slave ataupun master, hal ini dibuktikan dengan bisa memberikan notifikasi untuk melakukan pairing keperangkat lain, maupun perangkat lain tersebut yang melakukan pairing ke module Bluetooth HC-05. Untuk melakukan setting perangkat Bluetooth dibutuhkan perintah-perintah AT Command yang mana perintah AT Command tersebut akan di respon oleh perangkat Bluetooth jika modul Bluetooth tidak dalam keadaan terkoneksi dengan perangkat lain.

Keterangan pinout di atas adalah sebagai berikut:

1. EN fungsinya untuk mengaktifkan mode AT Command Setup pada modul HC-05. Jika pin ini ditekan sambil ditahan sebelum memberikan tegangan ke modul HC05, maka modul akan mengaktifkan mode AT Command Setup. Secara default, modul HC-05 aktif dalam mode Data.
2. Vcc adalah pin yang berfungsi sebagai input tegangan. Hubungkan pin ini dengan sumber tegangan 5V.
3. GND adalah pin yang berfungsi sebagai ground. Hubungkan pin ini dengan ground pada sumber tegangan.
4. TX adalah pin yang berfungsi untuk mengirimkan data dari modul ke perangkat lain (mikrokontroler). Tegangan sinyal pada pin ini adalah 3.3V sehingga dapat langsung dihubungkan dengan pin RX pada arduino karena tegangan sinyal 3.3V dianggap sebagai sinyal bernilai HIGH pada arduino.
5. RX adalah pin yang berfungsi untuk menerima data yang dikirim ke modul HC05. Tegangan sinyal pada pin sama dengan tegangan sinyal pada pin TX, yaitu 3.3V. Untuk keamanan, sebaiknya gunakan pembagi tegangan jika menghubungkan pin ini dengan mikrokontroller yang bekerja pada tegangan 5V.Pembagi tegangan tersebut menggunakan 2 buah resistor.Resistor yang digunakan sebagai pembagi tegangan pada tutorial ini adalah 1K ohm dan 2K ohm. Untuk lebih jelasnya, dapat dilihat pada bagian implementasi koneksi antara modul HC-05 dan mikrokontroller.

6. STATE adalah pin yang berfungsi untuk memberikan informasi apakah modul terhubung atau tidak dengan perangkat lain.

Mobile robot adalah robot yang mampu bergerak secara bebas karena memiliki alat gerak, sehingga mampu berpindah posisi. Secara umum mobile robot dibedakan oleh sistem penggerak (locamotion system). Locomotion system merupakan gerakan melintasi permukaan datar yang dapat disesuiakan dengan medan yang akan dilalui sesuai dengan tugas yang diberikan pada robot.



Gambar 36. Smart Robot Bluetooth HC-05

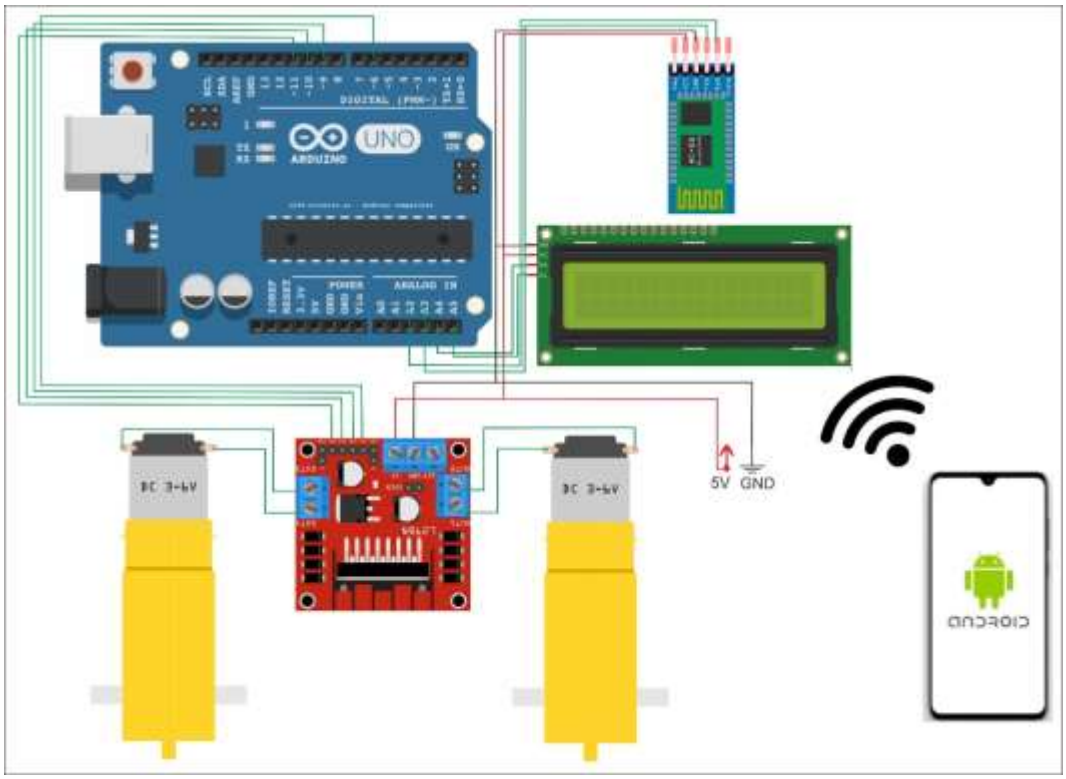
Robot memiliki banyak bentuk dan kendali seperti robot beroda dan berkaki, robot beroda adalah robot yang bergerak dengan menggunakan roda. Roda merupakan pembantu gerak, paling mudah dan paling efisien digunakan untuk menggerakkan robot melintasi permukaan datar. Roda banyak dipilih, karena mudah diperoleh, mudah dipakai dan memberikan traction yang bagus. Traction merupakan variabel yang terdapat pada roda dan permukaan yang dilintasi oleh roda. Material roda yang lebih lembut memiliki koefisien traction yang besar, sehingga mampu memberikan gesekan (friction) yang besar pula, dan menggunakan daya untuk menggerakan motor. Jumlah roda yang digunakan pada robot dapat beragam sesui kebutuhan.

C. Alat dan Bahan

1) Trainer Robotika 1 buah
2) Komputer/laptop 1 buah
3) Kabel USB Arduino 1 buah
4) Jumper Male-male 9 buah
5) Aplikasi Smart Car 1 buah
6) Smartphone 1 buah

D. Langkah Percobaan

1. Buatlah rangkaian robot mobile kontrol bluetooth dengan mengikuti skema berikut:



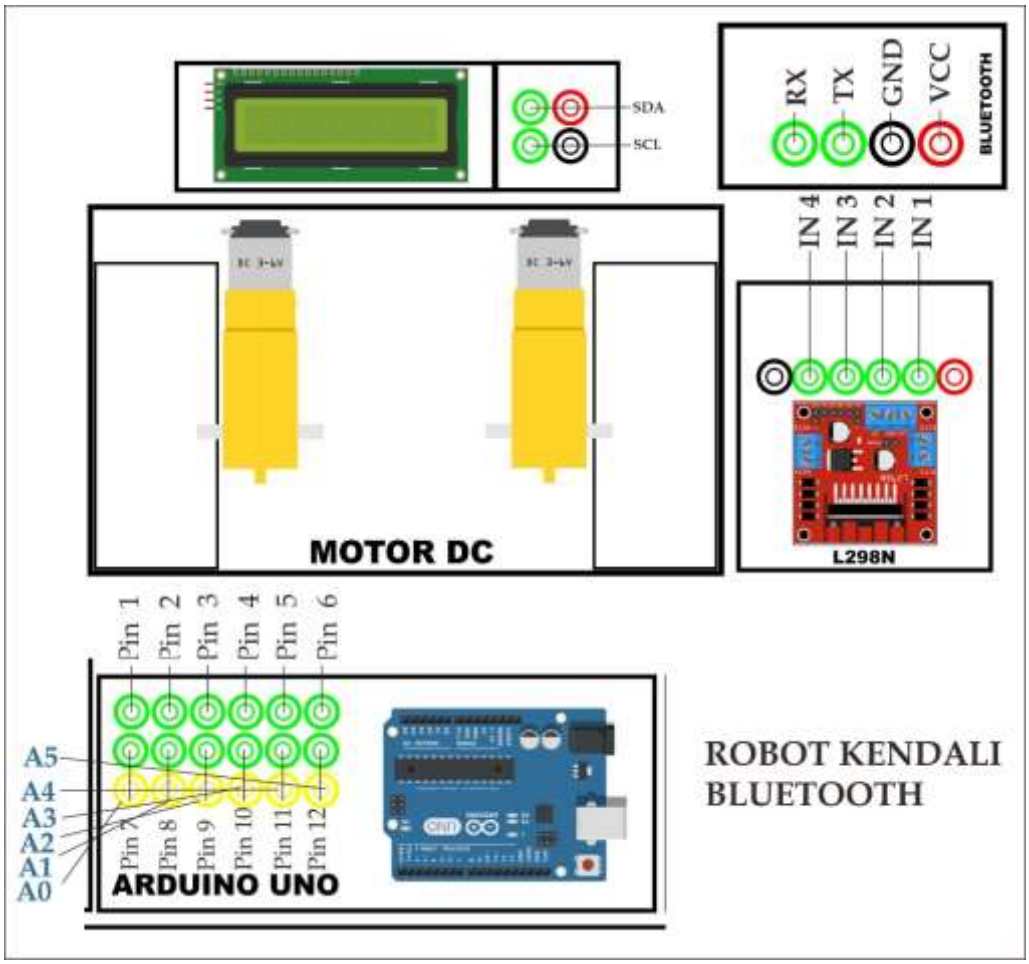
Gambar 37. Skema rangkaian robot manual kendali Bluetooth

Konfigurasi pin rangkaian robot manual kendali joystick

Arduino Uno	Antarmuka
A2	TX Bluetooth
A3	RX Bluetooth

A4	SDA i2C LCD
A5	SCL i2C LCD
D6	IN4 Motor DC
D9	IN3 Motor DC
D10	IN2 Motor DC
D11	IN1 Motor DC

Rangkaian tersebut dapat dipraktikkan pada trainer dengan menghubungkan pin A2 Arduino Uno ke TX Bluetooth, pin A3 Arduino Uno ke RX Bluetooth, pin A4 Arduino Uno ke SDA i2C LCD, pin A5 Arduino Uno ke SCL i2C LCD, pin D6 Arduino Nano ke IN4 Driver motor, pin D9 Arduino Nano ke IN3 Driver motor, pin D10 Arduino Nano ke IN2 Driver motor, pin D11 Arduino Nano ke IN1 Driver motor, kemudian Red to 5V , dan terakhir Black to Ground, seperti gambar pemasangan *jumper* pada trainer berikut ini:



Gambar 38. Rangkaian Robot manual kendali bluetooth

2. Buka aplikasi Arduino IDE di Komputer/Laptop dan masukkan Sketch program berikut:

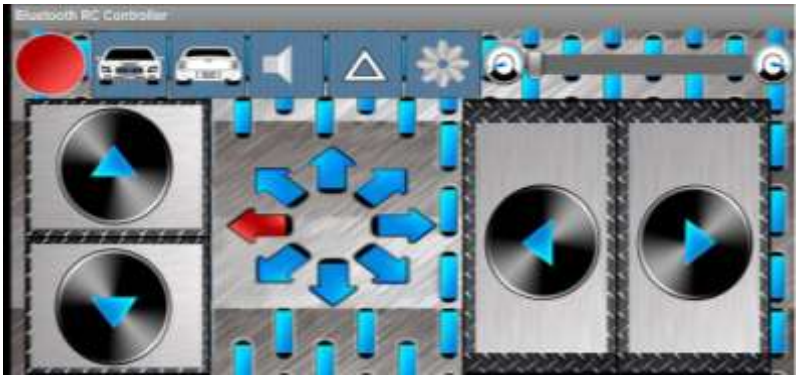
```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(A3, A2); // RX, TX
  #define m1 6
  #define m2 9
  #define m3 10
  #define m4 11
  int data=0,kec=0;
  boolean maju=true;
//array kecepatan
  int fast[11]={0,80,100,120,140,160,180,200,220,240,255};
void setup()
{
  pinMode(m1,OUTPUT);
  pinMode(m2,OUTPUT);
  pinMode(m3,OUTPUT);
  pinMode(m4,OUTPUT);
  // for HC-05 use 38400 when poerwing with KEY/STATE set to
HIGH on power on
  mySerial.begin(9600);
}
void motorOut(unsigned char lpwm, unsigned char rpwm, boolean
arrow){
  //arrow=false=maju; arrow=true=mundur;
  if(arrow==false){
  digitalWrite(m3,HIGH);
  digitalWrite(m1,LOW);
  analogWrite(m4,255-lpwm);
  analogWrite(m2,rpwm);
}
else{
  digitalWrite(m3,LOW);
  digitalWrite(m1,HIGH);
  analogWrite(m4,lpwm);
  analogWrite(m2,255-rpwm);
}
}

void loop(){
  if(mySerial.available()>0){
  data=mySerial.read();
  //penyimpan data kecepatan
  if (data =='0') { kec=0;}
  else if (data =='1') { kec=1;}
  else if (data =='2') { kec=2;}
  else if (data =='3') { kec=3;}
  else if (data =='4') { kec=4;}
  else if (data =='5') { kec=5;}
  else if (data =='6') { kec=6;}
  else if (data =='7') { kec=7;}
  else if (data =='8') { kec=8;}
  else if (data =='9') { kec=9;}
  else if (data =='q') {kec=10;}

//S= Stop
```

```
  if (data == 'S')
  { motorOut(0,0,false); }
//F=Maju
  If (data=='F')
  { motorOut(fast[kec],fast[kec],true); }
//I=Maju sambil belok kanan
  if (data=='I')
  { motorOut(fast[kec],((fast[kec])/2),true);}
//G=Maju sambil belok kiri
  if (data=='G')
  { motorOut(((fast[kec])/2),fast[kec],true); }
//R=Belok kanan
  if(data=='R')
  { motorOut(fast[kec],0,true); }
//L=Belok kiri
  if(data=='L')
  { motorOut(0,fast[kec],true); }
//B=Mundur
  if(data=='B')
  { motorOut(fast[kec],fast[kec],false); }
//H=Mundur kiri
  if (data=='H')
  { motorOut(((fast[kec])/2),fast[kec],false); }
//Mundur kanan
  if (data=='J')
  { motorOut(fast[kec],((fast[kec])/2),false); }
}
}
```

3. Compile program dengan menekan tombol Verify/Compile pada aplikasi Arduino IDE atau dengan menekan tombol Ctrl + R pada keyboard untuk memastikan program berhasil dan tidak terdapat kesalahan.
4. Hubungkan Laptop dengan Arduino Uno menggunakan Kabel USB Arduino
5. Upload program pada computer/laptop ke Arduino Uno dengan menekan tombol Upload pada aplikasi Arduino IDE atau dengan menekan tombol Ctrl + U pada keyboard.
6. Install aplikasi android yang telah disiapkan di barcode trainer:



Gambar 39. Aplikasi Android Smart Car

E. Hasil Percobaan

1. Kondisi awal *mobile robot* (Kontrol Bluetooth) dalam keadaan tidak bergerak ;
2. Maju pada aplikasi android Motor Kiri dan Motor kanan bergerak maju;
3. Kiri pada aplikasi android Motor Kiri berhenti/stop dan Motor kanan bergerak maju;
4. Kanan pada aplikasi android Motor kanan berhenti/stop dan Motor Kiri bergerak maju;
5. Mundur pada aplikasi android Motor Kiri dan Motor kanan bergerak mundur;

PERCOBAAN VII

ANTARMUKA MOTOR BRUSHLESS ARDUINO (ROBOT TERBANG)

PERCOBAAN VII | ANTARMUKA MOTOR BRUSHLESS ARDUINO (ROBOT TERBANG)

A. Tujuan Percobaan

1) Mahasiswa mampu menjelaskan prinsip kerja motor brushlesh.
2) Mahasiswa mampu mengimplementasikan motor brushlesh sebagai robot terbang pada rangkaian Trainer robotika.
3) Mahasiswa mampu melakukan pemograman motor brushlesh sebagai robot terbang pada Aplikasi Arduino IDE.

B. Teori Dasar

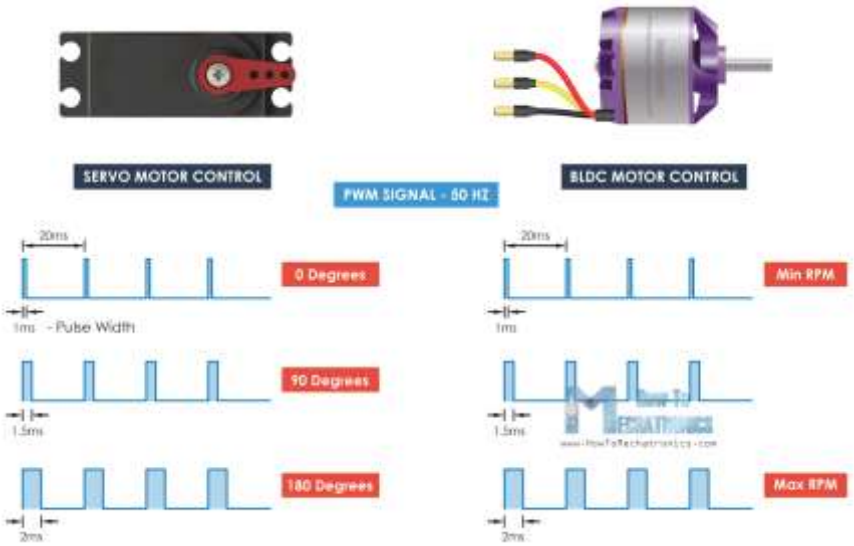
Robot terbang membutuhkan penggerak berupa baling-baling yang diputar oleh motor. Spesifikasi yang harus dipenuhi oleh sistem gerak ini adalah torsi, efisiensi, dan getaran yang ditimbulkan oleh berputarnya motor dan baling-baling. Motor dengan getaran yang terlalu besar dapat mengganggu sensor-sensor yang digunakan pada quadcopter. Efisiensi motor berkaitan dengan durabilitas terbang dari quadcopter, mengingat sumber data (battery) yang digunakan terbatas. Motor brushless memiliki beberapa kelebihan yaitu: efisiensi tinggi, kecepatan dan torsi yang tinggi, respons dinamis yang tinggi, dan masa operasi yang panjang.

Gambar 40. Motor Brushlesh dan ESC

Keuntungan dari brushless motor sebagai berikut:

- Komputer dapat mengatur kecepatan motor lebih baik sehingga membuat brushless motor lebih efisien.

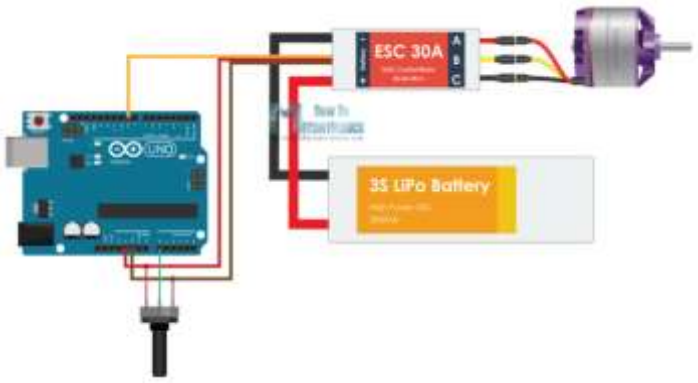
- Tidak adanya storing/electrical noise.
- Tidak menggunakan brushes yang dapat rusak setelah lamanya pemakaian.
- Dengan posisi electromagnets di bagian stator, maka pendinginan motor menjadi lebih mudah.
- Jumlah electromagnets di stator dapat sebanyak mungkin untuk mendapatkan kontrol yang lebih akurat.



Gambar 41. PWM Signal

ESC (Elektronic Speed Control) yang berfungsi sebagai pengatur kecepatan motor, selain itu juga berfungsi untuk menaikkan jumlah arus yang diperlukan oleh motor. ESC dapat dikatakan juga sebagai Drive motor dengan mengeluarkan pulsa untuk brushless motor yang berasal dari mikrokontroler.

Modul ini memiliki motor BLDC yang lebih cepat dengan spesifikasi sebagai berikut: memiliki peringkat KV 1000, dapat diberi daya menggunakan baterai 2S, 3S , 4S LiPo atau tegangan power supply dan membutuhkan 30A ESC.



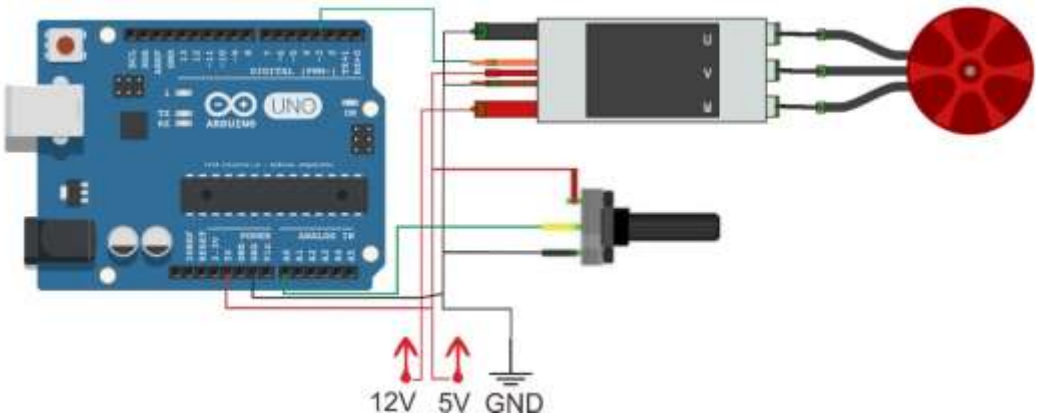
Gambar 42. Contoh rangkaian Ujicoba

C. Alat dan Bahan

1) Trainer Robotika 1 buah
2) Komputer/laptop 1 buah
3) Kabel USB Arduino 1 buah
4) Jumper Male-male 9 buah

D. Langkah Percobaan

1. Buatlah rangkaian motor brushless Arduino (robot terbang) dengan mengikuti skema berikut:

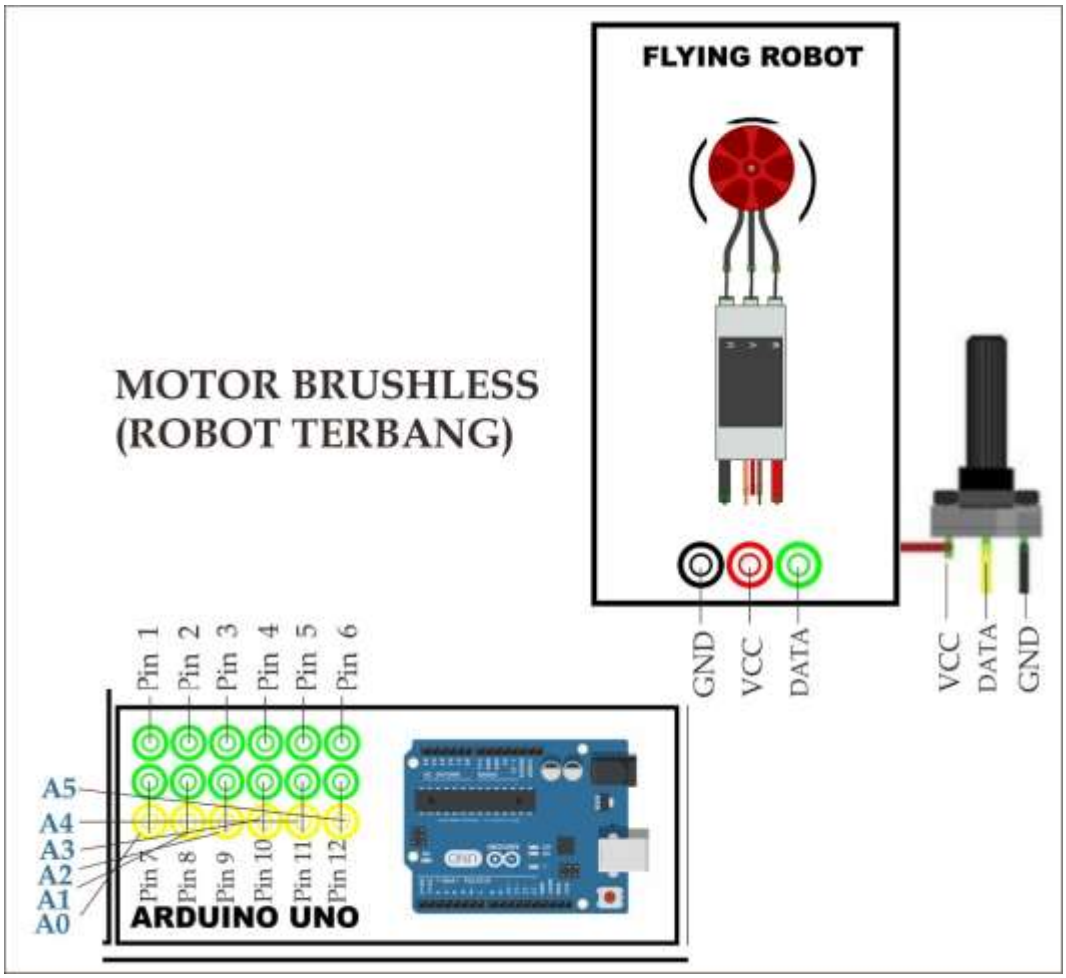


Gambar 43. Skema rangkaian motor brushless Arduino (robot terbang)

Konfigurasi pin rangkaian robot manual kendali joystick

Arduino Uno	Antarmuka
A0	Data Potensio
D3	Data ESC

Rangkaian tersebut dapat dipraktikkan pada trainer dengan menghubungkan pin A0 Arduino Uno ke Data Potensio, pin D3 Arduino Uno ke Data ESC, kemudian Red to 5V , dan terakhir Black to Ground, seperti gambar pemasangan *jumper* pada trainer berikut ini:



Gambar 44. Rangkaian motor brushless (robot terbang)

2. Buka aplikasi Arduino IDE di Komputer/Laptop dan masukkan Sketch program berikut:

```
#include <Servo.h>
Servo ESC;     // create servo object to control the ESC
int potValue;  // value from the analog pin
void setup() {
  // Attach the ESC on pin 3
  ESC.attach(3,1000,2000); // (pin, min pulse width, max pulse
width in
microseconds)
}
void loop() {
  potValue = analogRead(A0);   // reads the value of the
potentiometer (value between 0 and 1023)
```

```
  potValue = map(potValue, 0, 1023, 0, 180);   // scale it to
use it with the servo library (value between 0 and 180)
  ESC.write(potValue);    // Send the signal to the ESC
}
```

3. Compile program dengan menekan tombol Verify/Compile pada aplikasi Arduino IDE atau dengan menekan tombol Ctrl + R pada keyboard untuk memastikan program berhasil dan tidak terdapat kesalahan.
4. Hubungkan Laptop dengan Arduino Uno menggunakan Kabel USB Arduino
5. Upload program pada computer/laptop ke Arduino Uno dengan menekan tombol Upload pada aplikasi Arduino IDE atau dengan menekan tombol Ctrl + U pada keyboard.

E. Hasil Percobaan

Pada tegangan apapun (diuji antara 7.4v dan 12v) trainer tidak dapat mengubah kecepatan putaran meskipun tegangan x 1000 adalah rpms. Menjalankan kode yang akan memutar motor perlahan tetapi tidak tetap pada standar pada trainer.

PERCOBAAN VIII

ANTARMUKA ***INTERNET OF ROBOTIC THINGS***

PERCOBAAN VIII | ANTARMUKA *INTERNET OF ROBOTIC THINGS*

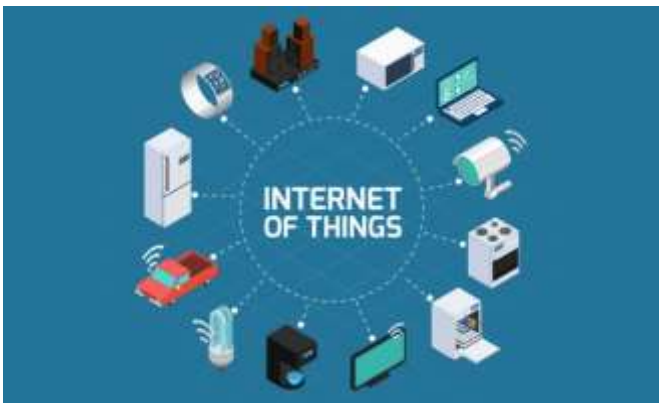
A. Tujuan Percobaan

1) Mahasiswa mampu menjelaskan prinsip kerja ESP 8266 sebagai penerapan internet pada robot.
2) Mahasiswa mampu mengimplementasikan ESP 8266 sebagai robot kendali jarak jauh pada rangkaian Trainer robotika.
3) Mahasiswa mampu melakukan pemograman ESP 8266 sebagai robot kendali jarak jauh pada Aplikasi Arduino IDE.

B. Teori Dasar

Internet of Things adalah suatu konsep dimana objek tertentu punya kemampuan untuk mentransfer data lewat jaringan tanpa memerlukan adanya interaksi dari manusia ke manusia ataupun dari manusia ke perangkat komputer.

Internet of Things leih sering disebut dengan singkatannya yaitu IoT. IoT ini sudah berkembang pesat mulai dari konvergensi teknologi nirkabel, micro-electromechanical systems (MEMS), dan juga Internet.IoT ini juga kerap diidentifikasikan dengan RFID sebagai metode komunikasi. Walaupun begitu, IoT juga bisa mencakup teknologi-teknologi sensor lainnya, semacam teknologi nirkabel maupun kode QR yang sering kita temukan di sekitar kita.



Gambar 45. Gambaran IoT

Apa saja kemampuan dari IoT? Adapun kemampuannya bermacam-macam contohnya dalam berbagi data, menjadi remote control, dan masih banyak lagi yang lainnya. Sebenarnya fungsinya termasuk juga diterapkan ke

benda yang ada di dunia nyata, di sekitar kita. Apa saja contohnya? Contohnya adalah untuk pengolahan bahan pangan, elektronik, dan berbagai mesin atau teknologi lainnya yang semuanya tersambung ke jaringan lokal maupun global lewat sensor yang tertanam dan selalu menyala aktif.

Jadi, sederhananya istilah Internet of Things ini mengacu pada mesin atau alat yang bisa diidentifikasikan sebagai representasi virtual dalam strukturnya yang berbasis Internet.

NodeMCU merupakan open source IOT platform yang ini termasuk firmware yang berjalan pada ESP8266 Wi-Fi SoC dari Espressif Systems, dan perangkat keras yang didasarkan pada modul ESP-12. Istilah "NodeMCU" secara default mengacu pada firmware daripada kit pengembangan.

Gambar 46. NodeMCU ESP 8266

NodeMCU dibuat tidak lama setelah ESP8266 keluar. Pada 30 Desember 2013, Espressif Systems memulai produksi ESP8266. ESP8266 adalah SoC Wi-Fi yang terintegrasi dengan inti Tensilica Xtensa LX106, Banyak digunakan dalam aplikasi IoT. NodeMCU dimulai pada 13 Oktober 2014, ketika Hong melakukan file nodemcu-firmware pertama ke GitHub. Dua bulan kemudian, proyek ini diperluas untuk mencakup sebuah platform terbuka-hardware ketika pengembang Huang R berkomitmen dalam Gerber file papan ESP8266, bernama devkit v0.9. Kemudian pada bulan itu, Tuan PM mengirim pustaka klien MQTT dari Contiki ke platform SoP ESP8266, dan berkomitmen untuk proyek NodeMCU, kemudian NodeMCU dapat mendukung protokol MQTT IoT, menggunakan Lua untuk mengakses broker MQTT. Pembaruan penting lainnya dilakukan pada 30 Januari 2015, ketika Devsaurus memindahkan port

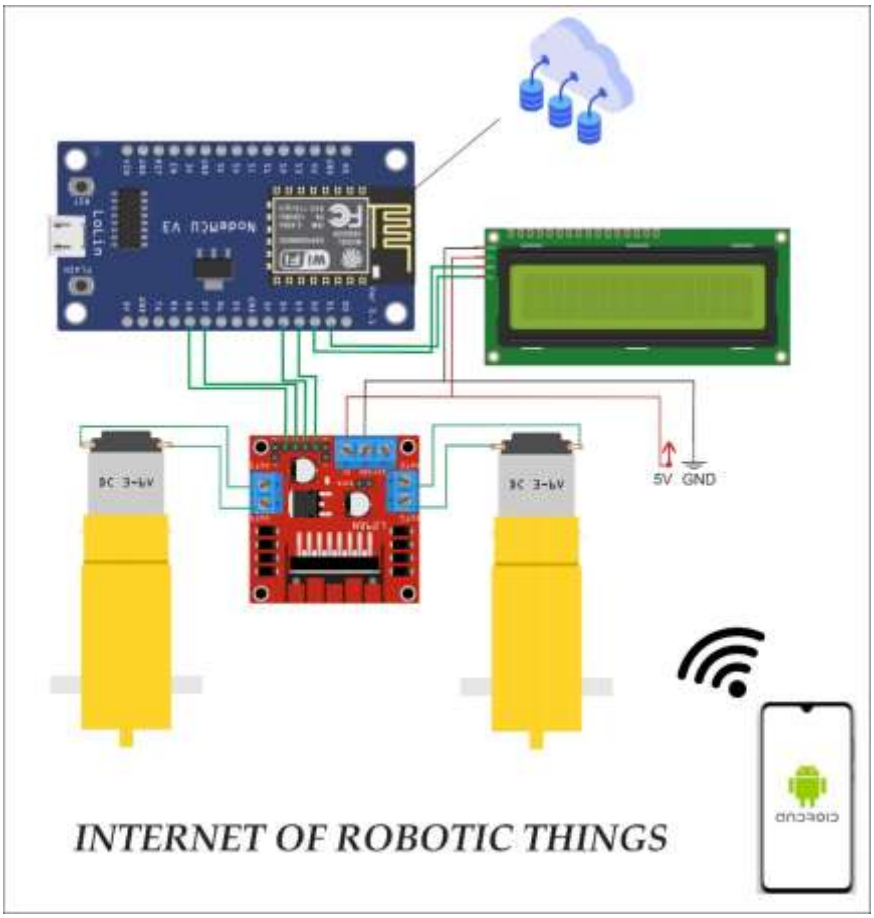
u8glib ke proyek NodeMCU, memungkinkan NodeMCU untuk dengan mudah mengarahkan layar LCD, Layar, OLED, bahkan VGA.

C. Alat dan Bahan

1) Trainer Robotika 1 buah
2) Komputer/laptop 1 buah
3) Kabel USB Arduino 1 buah
4) Jumper Male-male 9 buah
5) Smartphone 1 buah
6) Jaringan Internet
7) Aplikasi IoT

D. Langkah Percobaan

1. Buatlah rangkaian *Internet of Robotic Things* dengan mengikuti skema berikut:

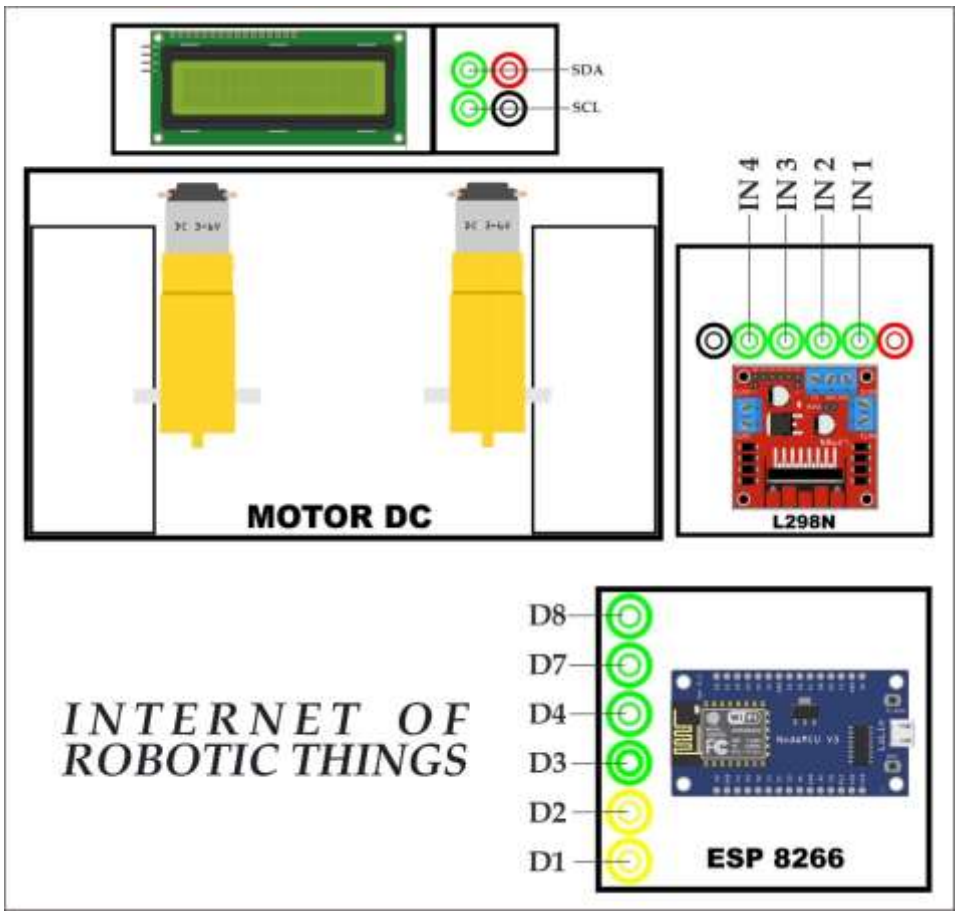


Gambar 47. Skema rangkaian *Internet of Robotic Things*

Konfigurasi pin rangkaian *Internet of Robotic Things*

NodeMCU ESP 8266	Antarmuka
D1	SCL i2C LCD
D2	SDA 12C LCD
D3	IN1 Driver Motor
D4	IN2 Driver Motor
D7	IN3 Driver Motor
D8	IN4 Driver Motor

Rangkaian tersebut dapat dipraktikkan pada trainer dengan menghubungkan pin D1 ESP 8266 ke SCL i2C LCD, pin D2 ESP 8266 ke SDA i2C LCD, pin D3 ESP 8266 ke IN1 Driver Motor, pin D4 ESP 8266 ke IN2 Driver Motor, pin D7 ESP 8266 ke IN3 Driver Motor, pin D8 ESP 8266 ke IN4 Driver Motor, kemudian Red to 5V , dan terakhir Black to Ground, seperti gambar pemasangan *jumper* pada trainer berikut ini:



Gambar 48. Rangkaian *Internet of Robotic Things*

2. Buka aplikasi Arduino IDE di Komputer/Laptop dan masukkan Sketch program berikut:

Program Tes Konektivitas :

```
#include <ESP8266WiFi.h>
WiFiClient client;
WiFiServer server(80);
const char* ssid = "YOUR SSID";      //enter your wi-fi user id
const char* password = "YOUR PASSWORD";   //enter your wi-fi
password
void setup()
{
  Serial.begin(115200);    //select the baud rate in serial
monitor
  connectWiFi();
  server.begin();
}
void loop()
{
}
void connectWiFi()
{
  Serial.println("Connecting to WIFI");
  WiFi.begin(ssid, password);
  while ((!(WiFi.status() == WL_CONNECTED)))
  {
    delay(300);
    Serial.print("..");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("NodeMCU Local IP is : ");     //Ip address that
you need to enter in the code
  Serial.print((WiFi.localIP()));
}
```

Program :

```
#include <ESP8266WiFi.h>
WiFiClient client;
WiFiServer server(80);
/* WIFI settings */
  const char* ssid = "YOUR SSID";   //enter your wi-fi user id
  const char* password = "YOUR PASSWORD";  //enter the wi-fi
password
/* data received from application */
String  data ="";
/* define L298N or L293D motor control pins */
  int leftMotorForward = 2;     /* GPIO2(D4) -> IN3   */
  int rightMotorForward = 15;   /* GPIO15(D8) -> IN1  */
  int leftMotorBackward = 0;    /* GPIO0(D3) -> IN4   */
  int rightMotorBackward = 13;  /* GPIO13(D7) -> IN2  */
/* define L298N or L293D enable pins */
  int rightMotorENB = 14; /* GPIO14(D5) -> Motor-A Enable */
  int leftMotorENB = 12;  /* GPIO12(D6) -> Motor-B Enable */

void setup()
```

```
{
  /* initialize motor control pins as output */
  pinMode(leftMotorForward, OUTPUT);
  pinMode(rightMotorForward, OUTPUT);
  pinMode(leftMotorBackward, OUTPUT);
  pinMode(rightMotorBackward, OUTPUT);
  /* initialize motor enable pins as output */
  pinMode(leftMotorENB, OUTPUT);
  pinMode(rightMotorENB, OUTPUT);
  /* start server communication */
  server.begin();
}

void loop()
{
    /* If the server available, run the "checkClient" function
*/
    client = server.available();
    if (!client) return;
    data = checkClient ();
/********************** Run function according to incoming data
from application *********************/
    /* If the incoming data is "forward", run the "MotorForward"
function */
    if (data == "forward") MotorForward();
    /* If the incoming data is "backward", run the "MotorBackward"
function */
    else if (data == "backward") MotorBackward();
    /* If the incoming data is "left", run the "TurnLeft" function
*/
    else if (data == "left") TurnLeft();
    /* If the incoming data is "right", run the "TurnRight"
function */
    else if (data == "right") TurnRight();
    /* If the incoming data is "stop", run the "MotorStop"
function */
    else if (data == "stop") MotorStop();
}
/********************************************* FORWARD
*************************************************/
void MotorForward(void)
{
  digitalWrite(leftMotorENB,HIGH);
  digitalWrite(rightMotorENB,HIGH);
  digitalWrite(leftMotorForward,HIGH);
  digitalWrite(rightMotorForward,HIGH);
  digitalWrite(leftMotorBackward,LOW);
  digitalWrite(rightMotorBackward,LOW);
}
/********************************************* BACKWARD
*************************************************/
void MotorBackward(void)
{
  digitalWrite(leftMotorENB,HIGH);
  digitalWrite(rightMotorENB,HIGH);
  digitalWrite(leftMotorBackward,HIGH);
  digitalWrite(rightMotorBackward,HIGH);
  digitalWrite(leftMotorForward,LOW);
  digitalWrite(rightMotorForward,LOW);
```

```
}
/********************************************* TURN LEFT
*****************************************************/
void TurnLeft(void)
{
  digitalWrite(leftMotorENB,HIGH);
  digitalWrite(rightMotorENB,HIGH);
  digitalWrite(leftMotorForward,LOW);
  digitalWrite(rightMotorForward,HIGH);
  digitalWrite(rightMotorBackward,LOW);
  digitalWrite(leftMotorBackward,HIGH);
}
/********************************************* TURN RIGHT
*****************************************************/
void TurnRight(void)
{
  digitalWrite(leftMotorENB,HIGH);
  digitalWrite(rightMotorENB,HIGH);
  digitalWrite(leftMotorForward,HIGH);
  digitalWrite(rightMotorForward,LOW);
  digitalWrite(rightMotorBackward,HIGH);
  digitalWrite(leftMotorBackward,LOW);
}
/********************************************* STOP
*****************************************************/
void MotorStop(void)
{
  digitalWrite(leftMotorENB,LOW);
  digitalWrite(rightMotorENB,LOW);
  digitalWrite(leftMotorForward,LOW);
  digitalWrite(leftMotorBackward,LOW);
  digitalWrite(rightMotorForward,LOW);
  digitalWrite(rightMotorBackward,LOW);
}
/********************************** RECEIVE DATA FROM the APP
*****************************************/
String checkClient (void)
{
  while(!client.available()) delay(1);
  String request = client.readStringUntil('\r');
  request.remove(0, 5);
  request.remove(request.length()-9,9);
  return request;
}
```

3. Compile program dengan menekan tombol Verify/Compile pada aplikasi Arduino IDE atau dengan menekan tombol Ctrl + R pada keyboard untuk memastikan program berhasil dan tidak terdapat kesalahan.
4. Hubungkan Laptop dengan NodeMCU ESP8266 menggunakan Kabel USB Arduino
5. Upload program pada computer/laptop ke NodeMCU ESP8266 dengan menekan tombol Upload pada aplikasi Arduino IDE atau dengan menekan tombol Ctrl + U pada keyboard.

6. Install aplikasi yg telah di siapkan di barcode:

Gambar 49. Aplikasi Robot IoT

E. Hasil Percobaan

1. Kondisi awal *mobile robot* (robot IOT) dalam keadaan tidak bergerak ;
2. Maju pada aplikasi android Motor Kiri dan Motor kanan bergerak maju;
3. Kiri pada aplikasi android Motor Kiri berhenti/stop dan Motor kanan bergerak maju;
4. Kanan pada aplikasi android Motor kanan berhenti/stop dan Motor Kiri bergerak maju;
5. Mundur pada aplikasi android Motor Kiri dan Motor kanan bergerak mundur;