

2020



DESAIN MOBILE ROBOT DENGAN KENDALI SMART PHONE ANDROID

Slamet Winardi
Arief Budijanto
Kunto Eko Susilo
Tresna Maulana Fahrudin



DESAIN *MOBILE* ROBOT DENGAN KENDALI *SMART PHONE* ANDROID

Author :

Slamet Winardi

Arief Budijanto

Kunto Eko Susilo

Tresna Maulana Fahrudin

Layouter :

Natasha AI

Editor :

Slamet Winardi, Arief Budijanto, Kunto Eko Susilo, Tresna Maulana Fahrudin

Design Cover :

Azizur Rachman

copyright © 2020

Penerbit



Scopindo Media Pustaka

Jl. Kebonsari Tengah No. 03, Surabaya

Telp. (031) 82519566

scopindomedia@gmail.com

Anggota IKAPI No. 241/JTI/2019

ISBN : 978-623-7729-85-3

Hak cipta dilindungi oleh Undang-undang Dilarang memperbanyak sebagian atau seluruh isi buku tanpa izin tertulis dari Penerbit

Sanksi Pelanggaran Pasal 113

Undang-undang Nomor 28 Tahun 2014 Tentang Hak Cipta

Setiap orang yang dengan atau tanpa hak melakukan pelanggaran terhadap hak ekonomi yang sebagaimana dimaksud dalam pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan ancaman pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp. 100.000.000 (seratus juta rupiah)

Setiap orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp. 500.000.000 (lima ratus juta rupiah).

Setiap orang dengan tanpa hak dan/atau tanpa izin Pencipta atau Pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk penggunaan Secara Komersial dipidana dengan pidana penjara paling lama (empat) tahun dan/atau pidana denda paling banyak Rp. 1.000.000.000 (satu miliar rupiah).



KATA PENGANTAR

Puja dan puji syukur kita panjatkan kepada Tuhan Yang Maha Esa yang selalu melindungi dan mendampingi umatnya dalam suka dan duka. Dalam masa wabah corona atau yang disebut *covid-19* memberi dampak kepada semua sector dan tak terhindarkan sektor pendidikan terkena imbas sehingga tidak dapat melaksanakan kegiatan belajar mengajar sebagaimana mestinya. Anjuran pemerintah untuk *Work From Home* (*#stayatbome*) dalam melaksanakan KBM harus ditaati.

Jadilah kegiatan Belajar Mengajar dilakukan secara *online* (daring) baik memberikan materi maupun tugas kepada mahasiswa juga dilakukan secara *live streaming* sehingga dapat bertatap muka kepada mahasiswa. Tentunya waktu yang ada menjadi lebih longgar karena tidak *full* mengurus administrasi kampus, sehingga perlu dimanfaatkan agar bermanfaat, timbulah ide untuk menulis sebuah buku.

Buku yang dibuat merupakan pengembangan dari sebuah materi pada saat memberikan pelatihan robotika bagi guru-guru SMK Petra dengan judul “DESAIN MOBILE ROBOT DENGAN KENDALI SMART PHONE ANDROID”. Semoga buku ini bermanfaat bagi siapa saja yang membaca dan dapat mempraktekan karena dibuat secara praktis dan mudah untuk melakukannya.

Tidak ada yang sempurna di dunia ini kecuali ciptaan Tuhan, Penulis mohon saran dan kritik untuk menyempurnakan buku ini, atas perhatiannya kami sampaikan terimakasih.

Surabaya, April 2020

Team Penulis





DAFTAR ISI

HALAMAN JUDUL.....	i
KATA PENGANTAR.....	iii
DAFTAR ISI.....	v
BAB I TEKNOLOGI <i>MOBILE</i> ROBOT	1
1.1 <i>Mobile</i> Robot	1
1.2 Arduino Nano	2
1.3 Modul <i>Bluetooth</i>	6
1.4 Chip IC L293.....	8
1.5 Software IDE Arduino	10
1.6 Software App Inventor.....	11
BAB II PERANGKAT KERAS <i>MOBILE</i> ROBOT	17
2.1 Modul <i>Bluetooth</i>	18
2.2 Pusat Pengendali.....	20
2.3 Antarmuka Motor DC	22
BAB PERANGKAT LUNAK <i>MOBILE</i> ROBOT.....	27
3.1 Implementasi Perangkat Lunak C Arduino.....	27
3.2 Implementasi Perangkat Lunak App Inventor.....	34
DAFTAR PUSTAKA	57
TENTANG PENULIS.....	59





BAB I

TEKNOLOGI *MOBILE* ROBOT

1.1. *Mobile Robot*

Ditinjau dari jenisnya robot terdiri dari dua jenis, yaitu *non mobile* dan *mobile robot*. Robot *non mobile* merupakan robot yang melaksanakan aksinya di suatu tempat atau tidak berpindah tempat, misalnya robot Manipulator Lengan yaitu merupakan robot yang di tempatkan khusus dan hanya mempunyai satu lengan yang berfungsi untuk memegang dan memindahkan barang dengan jarak yang tidak jauh, sedangkan *mobile robot* merupakan robot yang melakukan aktivitasnya dengan bergerak dan berpindah-pindah dari satu tempat ke tempat yang lain.

Mobile Robot merupakan konstruksi robot yang mempunyai ciri khas dari robot tersebut adalah aktuator yang berupa roda agar badan robot secara keseluruhan dapat digerakkan oleh roda tersebut, sehingga robot tersebut bisa perpindahan lokasi sesuai dengan perintah yang diberikan oleh kontroler.

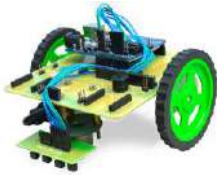
Mobile Robot ini banyak digemari oleh orang yang mulai belajar tentang robot. Membuat *mobile robot* tidak memerlukan tenaga yang ekstra keras karena fisik robot yang kecil dan ringan serta simple. Dalam pembuatan sebuah *mobile robot* diperlukan pengetahuan tentang alat kontrol yang dinamakan mikrokontroler, sensor-sensor elektronik dalam hal ini sensor garis atau sensor ultrasonic, dan actuator yaitu sepasang motor berikut driver penggeraknya sehingga robot dapat bergerak maju, mundur, maupun belok kanan atau kiri.

Kerangka mekanik *mobile robot* dapat dibuat dengan mudah menggunakan tripleks, akrilik atau menggunakan logam (aluminium).



Mobile robot banyak varian diantaranya pengikut garis (*Line Follower*), pengikut dinding (*Wall Follower*) atau pengikut cahaya (*Light Follower*).

Pengembangan *mobile robot* yang dilakukan oleh beberapa mahasiswa bahkan siswa-siswa SMA turut serta dalam mengembangkan *mobile robot* ini untuk beberapa fungsi, diantaranya adalah robot *line follower*, *maze solving* dan beberapa bentuk robot lain yang lebih unik seperti yang sekarang banyak dikembangkan oleh kegiatan ekstrakurikuler Robotik dan dalam beberapa perlombaan pun yang diadakan untuk menguji kemampuan anak-anak untuk menyerap dan mendalami ilmu tentang robotika. Berbagai model dan bentuk *mobile robot* yang dikembangkan oleh anak-anak bangsa ditunjukkan pada gambar 1.



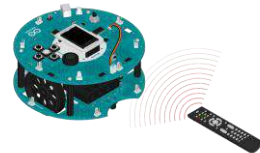
a. *Robot Line Follower*



b. *Robot Wall Follower*



b. *Obstacle Avoidance Robot*



d. Robot dikendalikan *remote controller*

Gambar 1. Macam-macam Mobile Robot

1.2. Arduino Nano

Arduino Nano merupakan *board* rangkaian yang dikembangkan oleh produsen mikrokontroler berbasis *chip ATmega328P* dengan ukuran yang relatif kecil namun secara fungsi tidak beda jauh dengan



pendahulunya yaitu Arduino Uno. Perbedaan dengan Arduino Uno terletak konektor Mini-B USB yang digunakan untuk komunikasi dengan komputer serta tidak mempunyai konektor power supply tetapi langsung terhubung dengan pin tegangan mikrokontroler 5 Volt.

Disebut sebagai *board* rangkaian pengembangan dari chip ATmega328P karena *board* ini memang berfungsi sebagai alat bantu dalam membuat *prototyping* rangkaian mikrokontroler. Dengan menggunakan *board* pengembangan ini, akan lebih mudah dalam merangkai rangkaian elektronika yang berbasis mikrokontroler jika dibanding dengan memulai merakit minimum sistem microcontroller ATmega328 dari awal. Bentuk fisik dari *board* pengembangan Arduino Nano tampak atas dan tampak bawah ditunjukkan pada gambar 2.



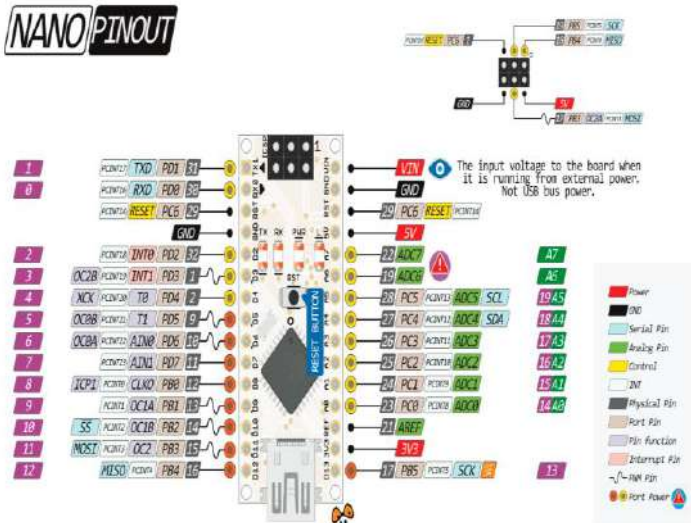
Gambar 2. Arduino Nano Development Board

Peta pin arduino ditunjukkan pada gambar 3. menjelaskan secara skematik dari *board* minimum system Arduino Nano. Pin Arduino Nano dijelaskan dalam 12 warna berbeda untuk menjelaskan fungsi masing-masing pin, warna merah digunakan untuk power supply baik tegangan 5 volt, 3,3 volt dan tegangan input (VIN) bila bekerja menggunakan tegangan dari luar, warna hitam pekat digunakan untuk pin ground (GND), warna biru muda (cyan) digunakan untuk pin serial yaitu standar RS232 (pin TX dan RX) pin serial standar I2C (pin SCL dan SDA) dan standar SPI (pin SS, MISO, MOSI, dan SCK), warna hijau digunakan untuk menjelaskan pin analog (ADC) yang terdiri dari pin ADC0 (pin A0 dalam Aduino) sampai dengan ADC7 (pin A7 dalam Arduino) serta pin Analog referensi (AREF), warna kuning sebagai pin



kontrol (RESET), warna putih sebagai pin interrupt (PCINT0 – PCINT5, PCINT8 – PCINT14, PCINT16 – PCINT23), hitam keabua-abuan untuk menandai pin fisik dari chip mikrokontroler ATmega328P, warna coklat untuk menandai pin port dari mikrokontroler (PB0 – PB5, PC0 – PC6, PD1 – PD7), warna ungu muda (*purple*) untuk menandai pin fungsi (T0, T1, AIN0, AIN1, CLKo, OC1A, OC1B, OC2), warna merah muda sebagai fungsi pin interrupt eksternal (INT0 dan INT1), warna ungu untuk menandai fungsi pin digital Arduino, dan pin yang bertanda gelombang sinus digunakan sebagai pin PWM atau DAC dari Arduino (3, 5, 6, 9, 10, dan 11).

Arus yang diijinkan untuk setiap pin Arduino Nano adalah 20mA dan maksimum dibatasi 40mA, bila ada arus yang melebihi 40mA disarankan menggunakan driver sehingga mikrokontroler aman terhadap *over current* dari rangkaian eksternal. Untuk mengisi program Arduino Nano ini menggunakan pin USB Jack mini tipe B dengan 5 pin. Jumlah total arus yang diperbolehkan untuk masuk pada pin yang ditandai bulatan merah dan kuning tidak diperbolehkan melebihi 100mA.



Gambar 3. Peta Pin Arduino Nano



Tabel 1. Spesifikasi Arduino Nano

Chip Mikrokontroler	ATMega328
Tegangan operasi	5V
Tegangan input	7V - 12V
Digital I/O pin	14 buah, 6 diantaranya menyediakan PWM
Analog Input pin	8 buah, 10 bit resolusi tiap pin analog
Arus DC per pin I/O	40 mA maksimal
Memori Flash	32 KB, 0.5 KB telah digunakan untuk <i>bootloader</i>
SRAM	2 KB
EEPROM	1 KB
Clock speed	16 Mhz
Dimensi	45 mm x 18 mm
Berat	5 g

Spesifikasi mikrokontroler Arduino Nano ditunjukkan dalam tabel di atas ini. Arduino Nano bekerja pada tegangan 5 volt atau tegangan eksternal melalui VIN sebesar 7 volt sampai dengan 12 volt, dengan pin input output digital sebanyak 14 pin (pin 0 – 13) ditambah pin yang dimultiplek dengan pin ADC sebanyak 6 pin (14 – 19 atau A0 – A5), untuk kebutuhan memori program sebanyak 32Kbyte yang dikurangi untuk boatloader sebesar 0.5 Kbyte, Static RAM yang tersedia sebesar 2 Kbyte, serta untuk memori data permanen menggunakan EEPROM sebesar 1 Kbyte dan bekerja dengan Kristal clock sebesar 16 MHz.



1.3. Modul *Bluetooth*

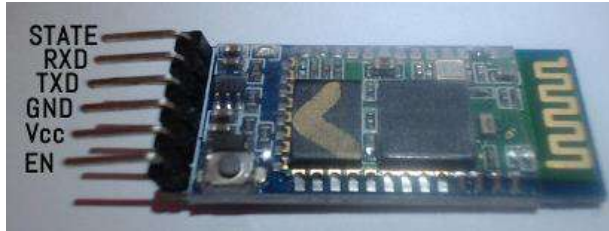
Bluetooth merupakan perangkat dengan spesifikasi industri untuk area jaringan-jaringan pribadi (*personal area networks* atau PAN) tanpa kabel. *Bluetooth* menghubungkan antar peralatan dan dapat dipakai untuk melakukan tukar-menukar informasi di antara peralatan-peralatan tersebut. Spesifikasi dari peralatan *Bluetooth* ini dikembangkan dan didistribusikan oleh kelompok yang tertarik dengan komunikasi tanpa kabel.

Bluetooth beroperasi dalam *band* frekuensi 2,4 GHz dengan menggunakan sebuah *frequency hopping traceiver* yang mampu menyediakan layanan komunikasi data dan suara secara *real-time* antara *host-host bluetooth* dengan jarak terbatas. Kelemahan teknologi ini adalah jangkauannya yang pendek dan kemampuan transfer data yang rendah.

Bluetooth berfungsi untuk media komunikasi antar perangkat sehingga mempermudah pengiriman atau *sharing file, audio* bahkan *video*. *Bluetooth* sendiri sebenarnya diciptakan untuk menggantikan media kabel sebagai media perantara sehingga lebih praktis dan efisien. Produk modul *bluetooth* yang sudah ada di pasaran salah satunya adalah *bluetooth HC-05*.

Modul *bluetooth HC-05* dapat melakukan komunikasi antara Arduino dan perangkat lain. Modul *bluetooth HC-05* adalah modul *bluetooth* yang dapat berfungsi sebagai *master* atau sebagai *slave*. Jika hanya ingin menggunakan Arduino sebagai *slave*, maka dapat menggunakan modul *bluetooth HC-06* karena modul tersebut secara default hanya dapat berfungsi sebagai *slave*. Modul HC-05 ini menggunakan *chipset* buatan *Cambridge Silicon Radio (CSR) BC417143* dan telah terpasang pada *breakout board*. Berikut ini adalah gambar modul *bluetooth HC-05* beserta keterangan pinoutnya.





Gambar 4. Bluetooth HC-05

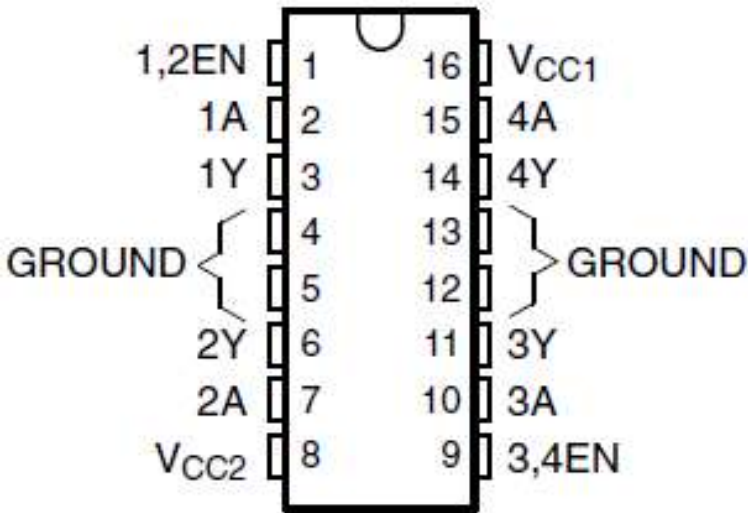
- **EN** fungsinya untuk mengaktifkan mode *AT Command Setup* pada modul HC-05. Jika pin ini ditekan sambil ditahan sebelum memberikan tegangan ke modul HC-05, maka modul akan mengaktifkan mode *AT Command Setup*. Secara default, modul HC-05 aktif dalam mode *Data*.
- **Vcc** adalah pin yang berfungsi sebagai input tegangan. Hubungkan pin ini dengan sumber tegangan 5V.
- **GND** adalah pin yang berfungsi sebagai *ground*. Hubungkan pin ini dengan *ground* pada sumber tegangan.
- **TX** adalah pin yang berfungsi untuk mengirimkan data dari modul ke perangkat lain (*microcontroller*). 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.
- **RX** adalah pin yang berfungsi untuk menerima data yang dikirim dari mikrokontroler ke modul HC-05. 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 Arduino 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 Arduino UNO.

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

1.4. Chip IC L293

Chip ini dirancang khusus sebagai penggerak motor DC. Konstruksi pin chip L293 dapat dilihat pada gambar 5. L293D adalah IC Driver Motor 16-pin yang dapat mengontrol satu set dua motor DC secara bersamaan ke segala arah. L293D dirancang untuk menyediakan arus penggerak dua arah hingga 600 mA (per saluran) pada tegangan dari 4,5 V hingga 36 V (pada pin 8). Anda dapat menggunakannya untuk mengendalikan motor DC kecil.

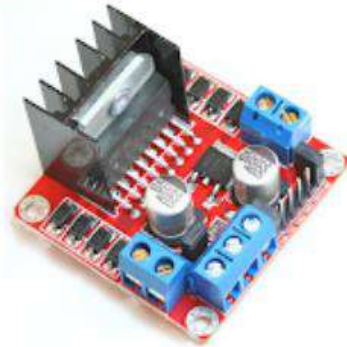


Gambar 5. Konstruksi pin chip L293D





Mini L293D



L298N Dual H Bridge

Gambar 6. Modul Driver Motor L293

Kedua modul ini berfungsi sama namun berbeda *package* IC nya sehingga bentuk modulnya pun berbeda, fungsi kerja dari driver motor tersebut dijelaskan dalam tabel di bawah ini. Chip L293 ini berisi dua driver sehingga satu chip dapat mengendalikan dua motor sekaligus, yang dipasang pada sisi kiri dan kanan dari sebuah robot. Pengaturan arah putaran motor menggunakan pin 1A, 2A dan 3A, 4A masing-masing untuk motor kanan dan kiri, pengaturan kecepatan dengan menggunakan pin *Enable* (EN12 dan EN34) dengan pengaturan berupa PWM. Kemudian saluran yang dihubungkan dengan motor DC menggunakan pin 1Y, 2Y dan 3Y, 4Y. Sumber tegangan untuk kedua motor menggunakan VCC1 dan VCC2.

Tabel 2. Fungsi pin chip L293

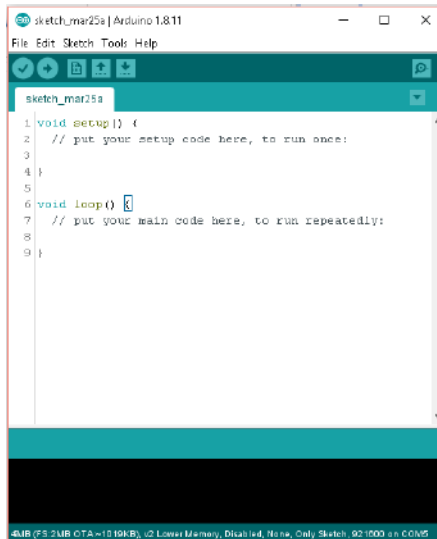
Pin	Deskripsi
Enable EN1,2 EN3,4	menerima perintah untuk menggerakkan dan mengatur kecepatan motor DC dengan memberikan sinyal digital high/low maupun sinyal PWM.
Input 1A,2A 3A,4A	Menerima input sinyal digital high/low untuk mengendalikan arah putaran motor DC.
Output 1Y,2Y 3Y,4Y	Saluran output masing-masing penggerak (<i>driver</i>) yang dihubungkan ke motor DC
VCC1 VCC2	Input sumber tegangan motor DC, dimana VCC1 untuk sumber tegangan rangkaian kontrol, VCC2 untuk sumber tegangan motorDC
Ground (GND)	Dihubungkan ke ground.

1.5. Software IDE Arduino

Kepanjangan dari IDE adalah *Integrated Development Environment*, atau bahasa Indonesia-nya adalah lingkungan pengembangan yang terintegrasi yang digunakan untuk penulisan suatu pemrograman. Dikenal sebagai *Environment* karena dengan software inilah pemrograman untuk Arduino dilakukan dengan membuat fungsi-fungsi melalui sintaks pemrograman yang akan diupload dalam chip Arduino. Dalam membuat program untuk Arduino digunakan bahasa pemrograman tersendiri yang menyerupai bahasa C yang sejatinya dibuat dari bahasa pemrograman *Java* dan merupakan turunan dari software *processing*. Bahasa pemrograman Arduino atau yang disebut dengan *Sketch* sudah dilakukan modifikasi dari software *processing* untuk memudahkan pengguna dalam membuat sebuah program dari bahasa aslinya. Dalam sebuah *board* Arduino, IC mikrokontroler ATmega328



telah diberi sebuah program *Bootlader* yang mempunyai fungsi sebagai penghubung antara komputer dengan board Arduino.



Gambar 7. Software IDE Arduino

Sebuah program yang diketik menggunakan *Software Arduino* (IDE) disebut dengan *sketch* (sketsa). *Sketch* yang ditulis dalam suatu IDE ini disimpan dalam sebuah file dengan ekstensi **.ino**. IDE pada *Software Arduino* memiliki beberapa fitur seperti *file* yang terdiri dari *new*, *open*, *save*, maupun *example* serta fitur *edit* yang terdiri dari *cutting/paste* dan *seraching/replacing* sehingga memudahkan dalam menulis kode program.

Pada IDE *Software Arduino* ini terdapat *message box* berwarna hitam di bawah yang berfungsi menampilkan status, seperti pesan *error*, *compile*, maupun *upload program* serta di bagian bawah paling kanan IDE *Software Arduino* ditunjukkan *board* yang digunakan serta *COM Ports* yang dipakai.

1.6. Software App Inventor

IDE (*Integrated Development Environment*) MIT App Inventor merupakan sebuah aplikasi pemrograman *visual block* dengan hanya *drag*



and drop programming yang memungkinkan kita untuk membuat program dan mengembangkan aplikasi berbasis Android tanpa harus mempunyai keahlian dan pengalaman dalam menggunakan *coding* bahasa pemrograman.

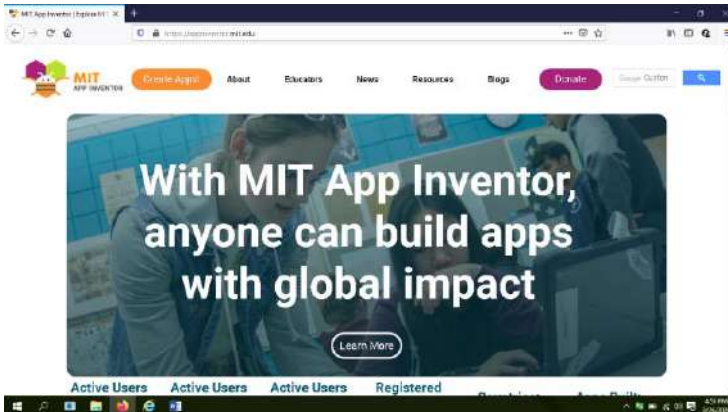
Dengan cukup mempunyai pemahaman alur logika dalam sebuah program, maka akan dapat membuat sebuah program aplikasi Android yang mudah dan sederhana.



Gambar 8. Logo MIT App Inventor

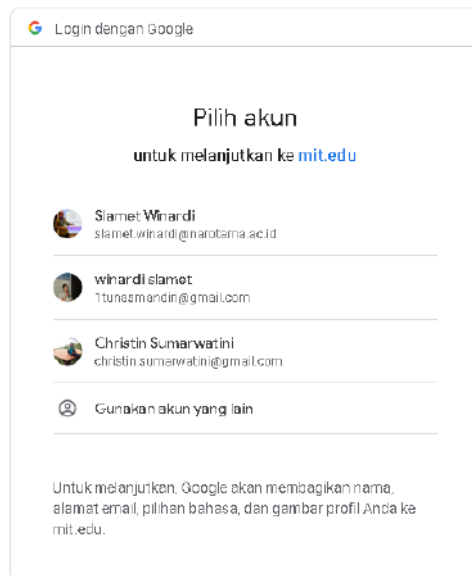
Visual Block Programming pada MIT App Inventor ini mampu mengubah bahasa pemrograman berbasis teks menjadi sebuah block *puzzle* yang dapat disusun sesuai dengan aplikasi yang akan dibuat. *App Inventor* yang dikelola oleh *Massachusetts Institute of Technology* (MIT) merupakan aplikasi *opensource* berbasis web yang disediakan oleh google. Untuk membuat sebuah project aplikasi Android di MIT App Inventor dimulai dengan mengakses alamat website <https://appinventor.mit.edu/>, setelah masuk ke *IDE MIT App Inventor*, selanjutnya klik **Create Apps!** untuk memulai membuat project baru.





Gambar 9. IDE MIT App Inventor

Setelah itu akan diarahkan ke login di *Google Account*, pilih akun google yang akan digunakan untuk masuk ke aplikasi App Inventor ini.

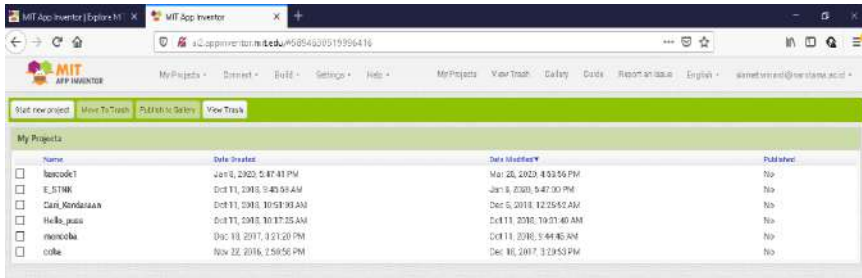


Afrikaans ▾ Bantuan Privasi Kebijakan

Gambar 10. Login Google Account



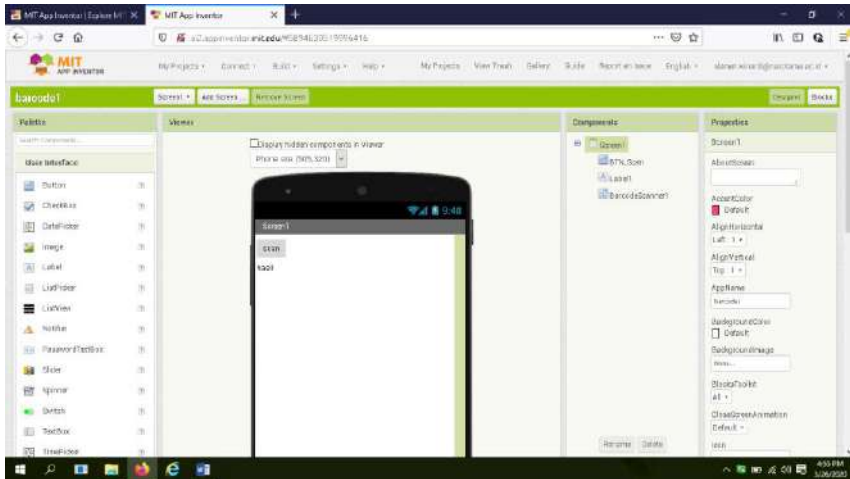
Kemudian akan diarahkan ke halaman *workspace project*, disitu akan ditunjukkan *list project* apa saja yang telah dibuat sebelumnya atau yang telah dimiliki. Bila ingin membuat project baru, pilih menu *Start new project* yang ada disamping kiri atas seperti yang ditunjukkan pada gambar dibawah ini. Ketikkan nama project yang akan dibuat, setelah itu klik OK untuk menyimpan.



Gambar 11. Membuat Project Baru App Inventor

Setelah itu akan diarahkan ke halaman *editor* aplikasi Android yang akan dibuat pada app inventor, pada samping kiri ada panel komponen *tools* yang bisa akan di *drag and drop* ke halaman aplikasi Android yang berada di tengah-tengah. Disamping sebelah kanan halaman desain adalah bagian *properties* untuk mengedit komponen yang sedang dipasang, maupun *workspace* dari Android yang sedang dibuat.





Gambar 12. *Workspace Android App Inventor*

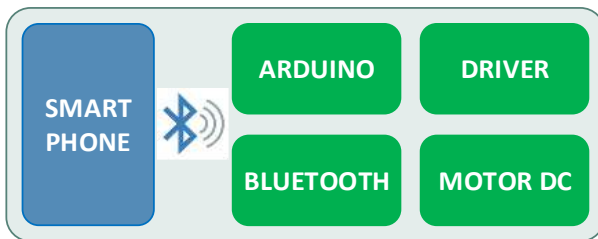




BAB II

PERANGKAT KERAS *MOBILE ROBOT*

Perangkat keras *mobile robot* yang dikendalikan melalui *Android Smart Phone* terdiri dari atas 3 bagian yaitu, Sensor, Pusat Pengendali, dan Aktuator. Pada bagian sensor dalam implementasinya menggunakan *bluetooth HC-05*, sedangkan pusat pengendalinya menggunakan modul Arduino Nano dan untuk aktuator dalam implementasinya menggunakan rangkaian penggerak motor DC menggunakan IC L293D. Diagram blok perangkat keras dari *mobile robot* ditunjukkan pada gambar 13.



Gambar 13. Diagram Blok Mobile Robot

Prinsip kerja *mobile robot* yang akan didesain adalah sebagai berikut:

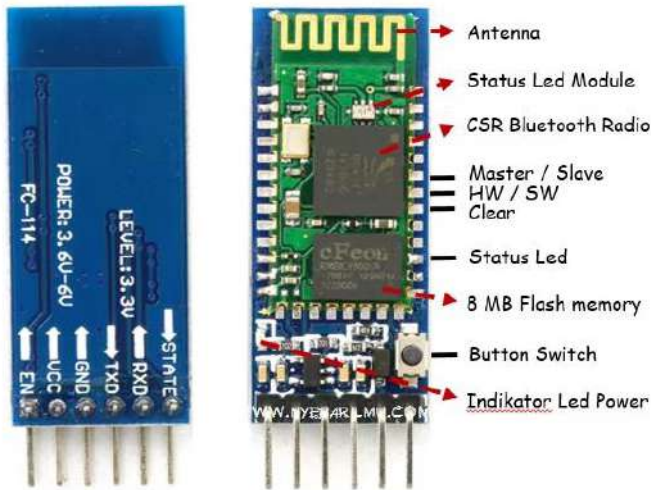
Robot akan dikendalikan menggunakan smart phone Android dengan perintah tombol-tombol maupun perintah suara. Tombol-tombol yang didesain di smart phone adalah tombol maju, mundur, kanan, kiri, dan stop, begitu juga untuk perintah suaranya. Jika tombol

ditekan maka akan mengirimkan sebuah kata yang dikirimkan melalui *Bluetooth* di smartphone dan dikirim ke penerima di robot dengan modul *Bluetooth* HC-05. *Bluetooth* akan meneruskan data ke mikrokontroler Arduino Nano dan akan diproses untuk dicocokkan dengan data yang ada di mikrokontroler, setelah sesuai dengan perintah yang diberikan maka mikrokontroler akan mengirimkan perintah ke *driver* motor L293 dan akan menjalankan sesuai dengan perintah yang diberikan oleh mikrokontroler, berikutnya perintah akan *latch* sehingga robot akan berjalan sesuai dengan perintah yang diberikan sampai perintah berikutnya diberikan.

2.1 Modul *Bluetooth*

Module *Bluetooth* HC-05 merupakan sebuah modul komunikasi tanpa kabel yang beroperasi pada frekuensi 2.4GHz dengan dua mode konektivitas yang dapat dipilih. Mode 1 berfungsi sebagai slave atau sebagai receiver data, mode 2 berfungsi sebagai master atau dapat berfungsi sebagai transceiver. Komunikasi yang dapat dipergunakan untuk mengakses modul ini yaitu serial dengan pin-pinnya TXD, RXD, VCC serta GND. Modul ini dilengkapi dengan LED (built in) sebagai indikator koneksi *bluetooth* terhadap perangkat lainnya seperti dengan smartphone android, ataupun dengan sesama modul *Bluetooth* dan sebagainya. Kemampuan modul ini mampu menjangkau radius 10 meter tanpa halangan, jika melebihi radius tersebut data yang dikirim atau diterima kurang maksimal. Modul ini memiliki 2 metode konfigurasi yaitu *AT Mode* dan *Communication Mode* dan dapat digunakan sebagai mode *slave* (Rx), maupun mode *master* (TX). Untuk *AT Mode* dapat difungsikan sebagai pengaturan konfigurasi dari modul *Bluetooth* HC-05, sedangkan pada *Communication Mode* berfungsi sebagai komunikasi tanpa kabel dengan perangkat lainnya.





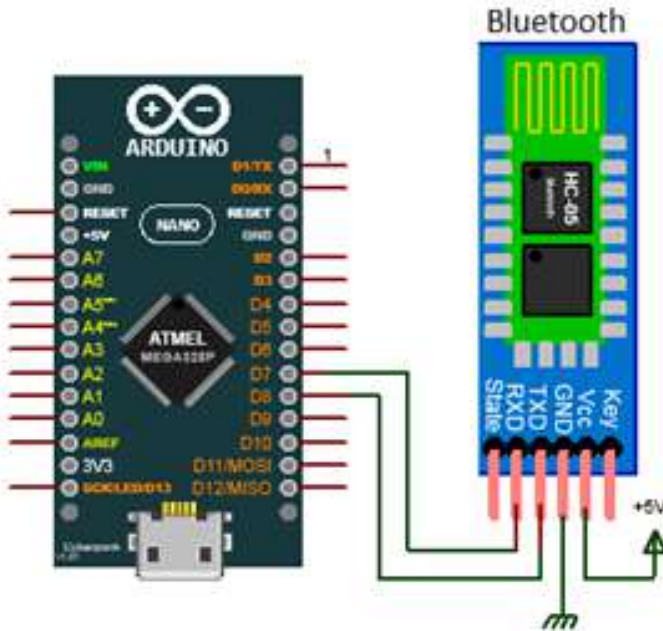
Gambar 14. Modul *Bluetooth* HC-05

Spesifikasi modul Bluetooth HC-05 :

- Bluetooth protocol : Bluetooth tipe v2.0+EDR
- Frekuensi kerja 2.4 GHz
- Baudrate 1Mbps pada mode sinkron
- Baudrate maksimum 2.1 Mbps / 160 kbps pada mode asinkron
- Tegangan antara 3,3 – 6 Volt DC
- Konsumsi arus rata-rata 50 mA
- *Modulasi Gaussian Frequency Shift Keying (GFSK)*
- Sensitivitas -84dBm (0.1% BER)
- Daya emisi 4 dBm
- Suhu operasional range -20°C — +75°C
- Keamanan dengan enkripsi data
- Dimensi modul 15.2×35.7×5.6 mm

Rangkaian antarmuka *Bluetooth* HC-05 dengan Arduino Nano ditunjukkan pada gambar 15. Pin TXD pada *Bluetooth* dihubungkan dengan pin D8 (yang difungsikan sebagai RX) dan pin RXD pada

Bluetooth dihubungkan dengan pin D7 (yang difungsikan sebagai TX), pin VCC dihubungkan dengan sumber tegangan (5 Volt) dan pin GND dihubungkan dengan *ground*. *Bluetooth* HC-05 berfungsi untuk menerima data teks yang dikirim dari *smart phone*. Data teks yang dikirimkan adalah “maju”, “mundur”, “kiri” dan “kanan”. Keluaran data dari *Bluetooth* akan dibaca oleh Arduino (pusat pengendali) secara serial. Kemudian data tersebut akan diolah lebih lanjut oleh pusat pengendali.



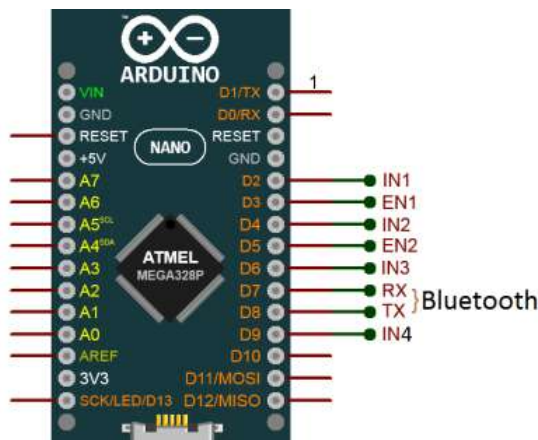
Gambar 15. Antarmuka Bluetooth Dengan Arduino Nano

2.2 Pusat Pengendali

Pusat pengendali *mobile robot* menggunakan *Arduino Nano*. Rangkaian pusat pengendali ini berfungsi untuk mengolah data teks yang dibaca dari *Bluetooth*. Data hasil pengolahan tersebut digunakan untuk mengatur kecepatan dan arah putaran motor DC. Pengaturan kecepatan motor DC dilakukan dengan cara membangkitkan sinyal *PWM* (*Pulse Width Modulation*) dari pin D3 dan pin D5. Sinyal *PWM*



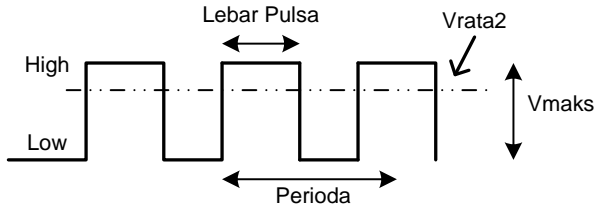
yang dibangkitkan dari pin D3 digunakan untuk mengendalikan kecepatan motor DC yang dihubungkan pada bagian roda robot sebelah kiri, sedangkan dari pin D5 untuk roda bagian sebelah kanan. Kemudian untuk pengaturan arah putaran motor DC (roda robot) sebelah kiri diatur melalui pin D2 dan pin D4, sedangkan untuk mengatur arah putaran motor DC sebelah kanan melalui pin D6 dan pin D9.



Gambar 16. Pusat Pengendali Mobile Robot

Sinyal PWM yang dibangkitkan dari pin D3 dan pin D5 bentuknya adalah gelombang kotak yang mempunyai amplitudo sekitar 5V dengan lebar pulsa yang bervariasi. Lebar pulsa berubah tergantung dari nilai PWM yang diberikan pada pin D3 dan pin D5. Nilai PWM tersebut berkisar 0 sampai dengan 255, karena nilai PWM yang dapat dibangkitkan dari Arduino Nano mempunyai resolusi 8 bit. Kemudian untuk menghitung tegangan rata-rata sinyal PWM dapat dilakukan dengan pendekatan praktis menggunakan rumus (1) dan (2).





Gambar 17. Bentuk Sinyal PWM

$$D = (\text{LebarPulsa} / \text{Perioda}) \times 100\% \quad \dots(1)$$

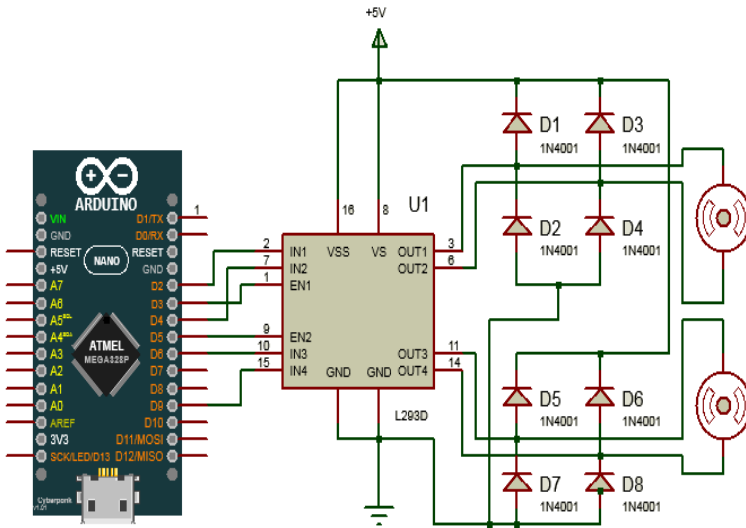
$$V_{rata2} = V_{maks} \times D \quad \dots(2)$$

Untuk memfungsikan pin mikrokontroler sebagai saluran masukan atau keluaran perlu didefinisikan dalam sebuah program. Jika pin difungsikan sebagai control maka pin tersebut harus diset sebagai output yang dilakukan pada subprogram ***void setup()*** kemudian fungsi khusus serial didefinisikan dalam perintah ***Serial.begin(baustrate)*** maka akan berfungsi sebagai komunikasi serial dengan device yang terpasang secara serial.

2.3 Antarmuka Motor DC

Rangkaian antar muka motor DC dengan *Arduino Nano* menggunakan *IC L293*. Dimana IC tersebut berfungsi sebagai penggerak motor DC yang dapat mengendalikan kecepatan dan arah putaran motor DC. Pengaturan kecepatan putaran motor DC dapat dilakukang dengan memberikan sinyal PWM pada input *IC L293*, yang mana sinyal *PWM* tersebut dibangkitkan dari output *Arduino Nano* pin D3 dan D9 yang diatur melalui program yang berada di dalam *chip ATmega328* yang merupakan chip mikrokontroler modul *Arduino Nano*. Gambar rangkaian antarmuka motor DC dengan *Arduino Nano* dapat lihat pada gambar 18.





Gambar 18. Antarmuka Penggerak Motor DC Dengan Arduino Nano

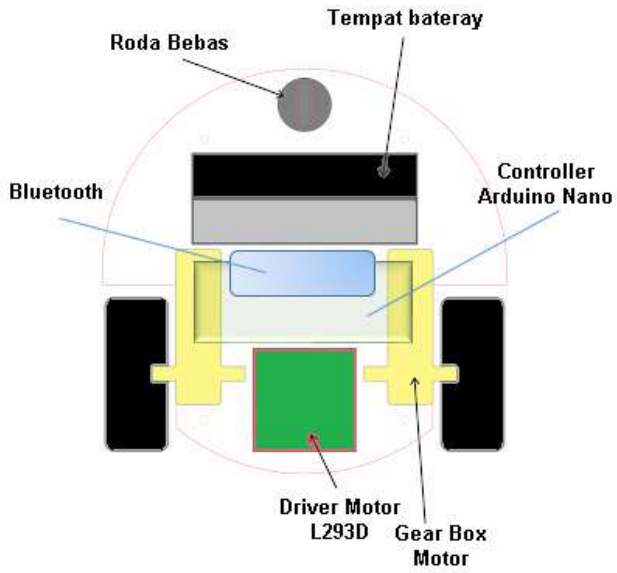
Untuk memprogram mikrokontroler yang menggerakkan driver motor sebagai actuator dari mobile robot tersebut maka perlu didefinisikan hubungan pin Arduino dengan pin driver motor L293D sehingga robot dapat dikontrol pergerakannya, robot dapat dikendalikan arah maju, mundur, belok kanan, belok kiri, maupun berhenti. Robot akan bergerak maju jika motor kanan dan motor kiri diberi perintah yang sama dan mempunyai *duty cycle* yang sama. Begitu juga untuk mundur perintahnya sama dengan maju tetapi kondisi control arahnya dibalik. Untuk belok kiri atau kanan maka salah satu roda robot akan dimatikan dengan memberikan nilai control pwm sebesar 0. Untuk menghentikan laju robot pin enable dari driver motor diberi nilai pwm sebesar 0 maka secara otomatis roda robot akan berhenti.

Tabel 3. Kondisi Logika Motor (Pin Arduino / Pin L293)

Motor Kanan			Motor Kiri			Keterangan
D2 / IN1	D4 / IN2	D3 / EN1	D2 / IN1	D4 / IN2	D3 / EN1	
IA1	IA2	ENA	IB1	IB2	ENB	Kode Software
HIGH	LOW	150	LOW	HIGH	0	Kiri
LOW	HIGH	150	LOW	HIGH	150	Maju
HIGH	LOW	150	HIGH	LOW	150	Mundur
LOW	HIGH	0	HIGH	LOW	150	Kanan
X	X	0	X	X	0	Berhenti

Ilustrasi mobile robot yang akan dibuat seperti gambar 19 berikut ini. Robot terdiri dari Sebuah mikrokontroler Arduino Nano, Sensor Bluetooth HC-05, Driver motor L293D untuk menggerakkan motor kanan dan motor kiri, yang selanjutnya akan menggerakkan roda. Tempat baterai sebagai sumber daya rangkaian control robot, dan untuk keseimbangan robot dipasang sebuah roda bebas. Robot akan diperintah menggunakan suara maupun tombol-tombol yang ada di smartphone. Bodi terbuat dari acrylic dengan tebal anatar 2 – 3 mm.





Gambar 19. Sketsa Mobile Robot





BAB III

PERANGKAT LUNAK

MOBILE ROBOT

Perangkat lunak yang digunakan untuk mengendalikan *mobile robot* terdiri dari 2 bagian yaitu perangkat lunak yang diterapkan pada modul *Arduino Microcontroller* menggunakan C Arduino dan yang diterapkan pada *smart phone* menggunakan *Android App Inventor*.

3.1 Implementasi Perangkat Lunak C Arduino

Implementasi perangkat lunak pada pusat pengendali *mobile robot* yaitu menggunakan C Arduino. Perangkat lunak tersebut terdiri dari *void setup()*; dan *void loop()*; Program dalam *void setup()*; merupakan inisialisasi dari komunikasi *Bluetooth* secara serial menggunakan pin 8 sebagai pengirim dan pin 7 sebagai penerima, yang implementasinya menggunakan *library SoftwareSerial.h* dengan deklarasi *SoftwareSerial BT(8,7)*; sedangkan untuk program inisialisasinya adalah *BT.begin(9600)*; dan inisialisasi pin-pin sebagai output yang disiapkan untuk mengendalikan motor DC. Program fungsi *void setup()*; ditunjukkan pada gambar 20.



```
#include <SoftwareSerial.h>

SoftwareSerial BT(8,7); //TX,RX
String readdata;

int IA1 = 2;
int IA2 = 4;
int ENA = 3;
int IB1 = 6;
int IB2 = 9;
int ENB = 5;

void setup() {
  BT.begin(9600);
  pinMode(IA1, OUTPUT);
  pinMode(IA2, OUTPUT);
  pinMode(ENA, OUTPUT);
  pinMode(IB1, OUTPUT);
  pinMode(IB2, OUTPUT);
  pinMode(ENB, OUTPUT);
}
```

Gambar 20. Program Fungsi *void setup()*

Keterangan :

#include <SoftwareSerial.h>

Pada *board* Arduino Nano hanya mempunyai satu pin RX dan TX, sehingga pada saat kondisi normal Arduino Nano hanya dapat melakukan komunikasi serial dengan satu perangkat tidak bisa lebih dari satu. Beberapa aplikasi ada yang mengharuskan microcontroller Arduino Nano untuk menjalankan komunikasi serial dengan beberapa perangkat dalam satu sistem. Untuk memfasilitasi hal tersebut, *board* Arduino Nano dilengkapi dengan fitur perangkat lunak yang dapat menjadikan pin Arduino Nano yang lain (pin 0 dan pin 1) sebagai pin RX-TX. Misalkan dalam sistem *mobile robot* ini menggunakan pin digital 7 sebagai RX dan pin digital 8 sebagai TX. Untuk dapat melakukan ini *library SoftwareSerial* dapat digunakan untuk mendefinisikan pin serial.

SoftwareSerial BT(8,7); //TX,RX

Bila ingin menambah port serial pada Arduino Nano dapat merubah pin I/O yg lain menjadi *Port serial* tambahan dengan menggunakan *class SoftwareSerial* . Dengan menambahkan *file header*



SoftwareSerial.h pada program dan membuat 1 buah objek *SoftwareSerial* untuk menambahkan 1 port serial, pada *mobile robot* yang dibuat 1 objek *softwareSerial* dengan nama BT.

```
int IA1 = 2;
int IA2 = 4;
int ENA = 3;
int IB1 = 6;
int IB2 = 9;
int ENB = 5;
```

memberi nama variable untuk setiap pin Arduino Nano yang digunakan dalam *mobile robot*. Deklarasi *int IA1=2; int IA2=4; int ENA=3;* adalah deklarasi pin Arduino Nano untuk mengendalikan motor DC sebelah kiri dan deklarasi *int IB1=6; int IB2=9; int ENB =5;* untuk mengendalikan motor DC sebelah kanan.

```
void setup() {
  BT.begin(9600);
  pinMode(IA1, OUTPUT);   pinMode(IB1, OUTPUT);
  pinMode(IA2, OUTPUT);   pinMode(IB2, OUTPUT);
  pinMode(ENA, OUTPUT);   pinMode(ENB, OUTPUT);
}
```

Void setup ini berfungsi untuk mendeklarasikan fungsi masing-masing pin yang digunakan dalam *mobile robot*. *Bluetooth* akan berkomunikasi dengan Arduino Nano menggunakan *Baudrate* (kecepatan transfer data di komunikasi serial) sebesar 9600 bps (*byte per second*). Perintah *pinMode* akan memberikan fungsi pin Arduino Nano, jika parameter yang disertakan adalah OUTPUT maka pin-pin tersebut akan berfungsi sebagai *output* (keluaran) artinya data dikirim dari Arduino Nano ke driver motor.



```

void loop() {
  while (BT.available())
  delay(10);
  char c = BT.read();
  readdata += c;
}
if (readdata.length() > 0){
  if (readdata == "maju")
  {
    Robot_Maju();delay(100);
  }
  else if(readdata== "mundur")
  {
    Robot_Mundur();delay(100);
  }
  else if (readdata=="kanan")
  {
    Robot_Belok_Kanan();delay(100);
  }
  else if (readdata == "kiri")
  {
    Robot_Belok_Kiri();delay(100);
  }
  else if (readdata == "stop")
  {
    Robot_Berhenti();delay(100);
  }
  readdata="";
}
}

```

Gambar 21. Program Fungsi *void loop()*;

Keterangan :

`while (BT.available())`

Perintah ini akan menunggu data yang dikirimkan oleh *smartphone* melalui komunikasi *bluetooth*, jika ada data yang diterima oleh Arduino Nano yang diterima melalui *bluetooth* maka blok program yang ada dalam perintah *while* akan dilakukan sampai data yang terakhir diterima.

`delay(10);`

Perintah ini akan menunda selama 10 milidetik sebelum perintah berikutnya dijalankan.

`char c = BT.read();`



Dalam program utama *void loop()* akan melakukan pembacaan data karakter yang dikirimkan dari *mobile phone* melalui *bluetooth* menggunakan perintah *char c = BT.read();*

```
readdata += c;
```

Perintah ini akan merangkai huruf demi huruf yang diterima oleh variabel *c* sampai membentuk sebuah kata, yang digunakan untuk perintah robot. Teks yang dikirimkan dan dirangkai dalam variabel **readdata** adalah “maju“, “mundur”, “kiri”, “kanan” dan “stop”. Data teks tersebut digunakan untuk pengambilan keputusan menjalankan arah gerakan *mobile robot*.

```
if (readdata.length() > 0)
```

Panjang data yang telah dirangkai jika melebihi dari teks kosong maka instruksi ini akan dijalankan, yang didalamnya berupa pencocokan data dari variabel **readdata** dengan data yang telah disiapkan untuk memerintahkan robot.

```
    if (readdata == "maju")
    {
        Robot_Maju();delay(100);
    }
```

Perintah berikut akan memanggil subprogram *Robot_Maju()* jika variabel **readdata** sama dengan kata “maju”.

```
    else if (readdata == "mundur")
    {
        Robot_Mundur();delay(100);
    }
```

Perintah berikut akan memanggil subprogram *Robot_Mundur()* jika variabel **readdata** sama dengan kata “mundur”.



```
else if (readdata== "kanan")
{
    Robot_Belok_Kanan();delay(100);
}
```

Perintah berikut akan memanggil subprogram Robot_Belok_Kanan() jika variabel **readdata** sama dengan kata “kanan”.

```
else if (readdata == "kiri")
{
    Robot_Belok_Kiri();delay(100);
}
```

Perintah berikut akan memanggil subprogram Robot_Belok_Kiri() jika variabel **readdata** sama dengan kata “kiri”.

```
else if (readdata == "stop")
{
    Robot_Berhenti();delay(100);
}
```

Perintah berikut akan memanggil subprogram Robot_Berhenti() jika variabel **readdata** sama dengan kata “stop”.

```
readdata="";
```

Perintah ini akan kembali mengosongkan variabel **readdata** untuk kemudian diproses isi ulang saat menerima data dari *bluetooth*.

Dalam melaksanakan gerakan *mobile robot* arduino akan mengeksekusi program fungsi sebagai berikut :

- Robot agar bergerak maju maka program fungsi yang dieksekusi adalah *void Robot_Maju()*. Setelah melakukan eksekusi program fungsi tersebut



- Robot agar bergerak maju maka program fungsi yang dieksekusi adalah *void Robot_Mundur()*.

```
void Robot_Maju()
{
  digitalWrite(IB1,LOW);
  digitalWrite(IB2,HIGH);
  analogWrite(ENB,150);
  digitalWrite(IA1,LOW);
  digitalWrite(IA2,HIGH);
  analogWrite(ENA,150);
}
```

Gambar 22.

Program Fungsi Robot Maju

```
void Robot_Mundur()
{
  digitalWrite(IB1,HIGH);
  digitalWrite(IB2,LOW);
  analogWrite(ENB,150);
  digitalWrite(IA1,HIGH);
  digitalWrite(IA2,LOW);
  analogWrite(ENA,150);
}
```

Gambar 23.

Program Fungsi Robot Mundur

Sesuai dengan tabel 3 yang direncanakan kondisi logika driver motor dengan pengaturan pwm (*pulse width modulation*) sebesar 150.

- Robot agar bergerak mundur maka program fungsi yang dieksekusi adalah *void Robot_Belok_Kanan()*.
- Robot agar bergerak maju maka program fungsi yang dieksekusi adalah *void Robot_Belok_Kiri()*.

```
void Robot_Belok_Kanan()
{
  digitalWrite(IB1,LOW);
  digitalWrite(IB2,HIGH);
  analogWrite(ENB,150);
  digitalWrite(IA1,HIGH);
  digitalWrite(IA2,LOW);
  analogWrite(ENA,0);
}
```

Gambar 24.

Fungsi Robot Belok Kanan

```
void Robot_Belok_Kiri()
{
  digitalWrite(IB1,HIGH);
  digitalWrite(IB2,LOW);
  analogWrite(ENB,0);
  digitalWrite(IA1,LOW);
  digitalWrite(IA2,HIGH);
  analogWrite(ENA,150);
}
```

Gambar 25.

Fungsi Robot Belok Kiri



Untuk mengarahkan pergerakan robot ke arah kanan atau kiri dengan cara menghentikan salah satu roda robot yaitu memberikan nilai pwm (*pulse width modulation*) sebesar 0.

- Robot agar berhenti maka program fungsi yang dieksekusi adalah *void Robot_Berhenti()* yaitu dengan memberikan nilai pwm (*pulse width modulation*) kepada semua driver motor kanan dan motor kiri dengan nilai sebesar 0.

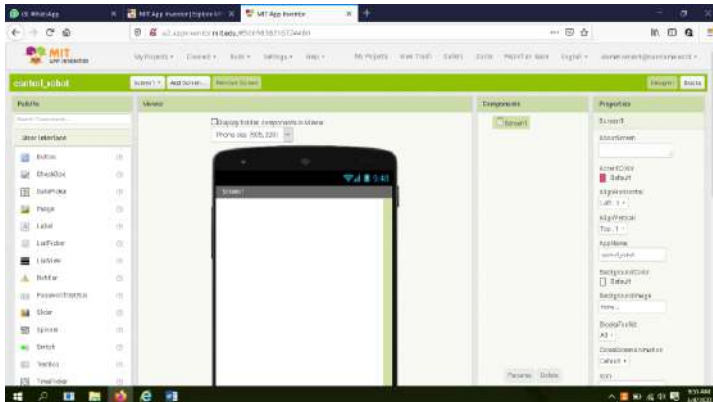
```
void Robot_Berhenti()
{
    digitalWrite(IB1, LOW);
    digitalWrite(IB2, HIGH);
    analogWrite(ENB, 0);
    digitalWrite(IA1, LOW);
    digitalWrite(IA2, HIGH);
    analogWrite(ENA, 0);
}
```

Gambar 26. Fungsi Robot Berhenti

3.2 Implementasi Perangkat Lunak App Inventor

Pembuatan aplikasi android untuk mengendalikan *mobile robot* melalui smart phone menggunakan MIT App inventor melalui layar sentuh. Pertama-tama buat project untuk control robot dengan mengklik menu my projects dan ikuti petunjuk selanjutnya.





Gambar 27. IDE MIT APP Inventor

Untuk membuat tombol-tombol perintah pilih **button** pada *user interface* dan kemudian di *drag and drop* pada *screen*.


Langkah-langkah Desain Kontrol Mobile Robot dengan Smartphone Android dengan software MIT App Inventor :

- ❖ Setelah IDE MIT App Inventor muncul kemudian kita melakukan *layout screen* yang digunakan sebagai *interface* kontrol, ganti **Screen1.title** = “Kontrol Mobile Robot”.
- ❖ *Drag nd drop* label untuk menampilkan status koneksi Bluetooth “*connected*” atau “*not connected*”.
- ❖ Klik **Pallette** kemudian pilih **layout** di bawah user interface selanjutnya pilih **HorizontalArrangement**, atur *properties* sebagai berikut :
 - AlignHorizontal = left
 - AlignVertical = top
 - Height = Automatic
 - Width = fill parent



❖ Selanjutnya pilih **Listpicker**, kemudian *drag and drop* ke **HorizontalArrangement1** kemudian atur *properties* sebagai berikut:

○ **Listpicker 1**


- *Hight* = *Automatic*
- *Width* = *fill parent*
- *image* = *bt2.jpg* 
- *TextAlignment* = *center*

❖ Pada layout ambil **TableArrangement** isikan lima buah tombol dengan pengaturan *properties* sebagai berikut :


○ **TableArrangement1**

- *Columns* = 3
- *Height* = *Automatic*
- *Width* = *fill parent*
- *Rows* = 3


○ **Button1**

- *Height* = *Automatic*
- *Width* = *Automatic*
- *Image* = *forward.jpg* 
- *Text* = "" (kosong)

○ **Button2**

- *Height* = *Automatic*
- *Width* = *Automatic*
- *Image* = *left.jpg* 
- *Text* = "" (kosong)

○ **Button3**

- *Height* = *Automatic*
- *Width* = *Automatic*
- *Image* = *right.jpg* 
- *Text* = "" (kosong)

○ **Button4**

- *Height* = *Automatic*



- *Width = Automatic*
- *Image = down.jpg*
- *Text = "" (kosong)*



○ **Button5**

- *Height = Automatic*
- *Width = Automatic*
- *Image = stop.jpg*
- *Text = "" (kosong)*



❖ Drag and drop layout **HorizontalArrangement** isikan empat buah **image** dan sebuah **label** dengan pengaturan properties sebagai berikut :

○ **HorizontalArrangement2**

- *AlignHorizontal = left*
- *AlignVertical = center*
- *Height = 100 pixel*
- *Width = fill parent*

○ **Image1 (Robot_atas)**

- *Height = fill parent*
- *Width = 30 persen*
- *Picture = robotmaju.png*
- *Visible = false*



○ **Image2 (Robot_bawah)**

- *Height = fill parent*
- *Width = 30 persen*
- *Picture = robotmundur.png*
- *Visible = false*





- ***Image3 (Robot_kiri)***
 - Height = fill parent
 - Width = 30 persen
 - *Picture = robotkiri.png*
 - *Visible = false*




- ***Image4 (Robot_kanan)***
 - Height = fill parent
 - Width = 30 persen
 - *Picture = robotkanan.png*
 - *Visible = false*



- ***Label2***
 - *FontBold = √*
 - *FontItalic = √*
 - *Fontsize = 20*
 - *FontTypeface = sans serif*
 - *Text = "" (kosong)*



- ❖ Drag and drop layout ***HorizontalArrangement*** isikan sebuah tombol dengan pengaturan properties sebagai berikut :
 - ***HorizontalArrangement2***
 - *AlignHorizontal* = right
 - *AlignVertical* = center
 - *Height* = fill parent
 - *Width* = 30 persen
 - ***Button6***
 - *BackgroundColor* = Default
 - *Height* = fill parent
 - *Width* = fill parent
 - *Image* = icon Mic 
 - *TextColor* = none

- ❖ Ambil komponen ***Bluetooth*** pada pallete → ***connectivity*** kemudian ambil komponen ***bluetoothclient***.
- ❖ Selanjutnya ambil komponen ***clock*** pada pallete → ***sensor*** kemudian ambil komponen clock yang mirip dengan gambar jam.
- ❖ Selanjutnya drag and drop komponen ***media*** yaitu ***SpeechRecognizer1***

Bentuk desain user interface smartphone kontrol *mobile robot* yang diharapkan seperti ditunjukkan pada gambar di bawah ini.





Gambar 28. User Interface Kontrol Mobile Robot

Setelah desain *interface* jadi berikutnya adalah mendesain *block* program untuk masing-masing komponen :

- ☞ Pindahkan dari menu design ke menu blocks dan hasil tampilannya sebagai berikut.



Gambar 29. Tampilan *Viewer Blocks Program*



- ☞ Klik **blocks** program kemudian pilih *pallette Listpicker1* kemudian pilih *block beforPicking* kemudian drag and drop pada *form viewer*.

```
when ListPicker1 .BeforePicking
do
```

Ketika sebelum diklik komponen *Listpicker1* komponen akan menampilkan beberapa device Bluetooth yang ada di sekitar, untuk menampilkan *set listpicker* sebagai berikut.

```
when ListPicker1 .BeforePicking
do set ListPicker1 .Elements to
```

Device Bluetooth akan ditampilkan pada listpicker jika komponen ini diklik, untuk desainnya disertakan sebagai berikut

```
when ListPicker1 .BeforePicking
do set ListPicker1 .Elements to BluetoothClient1 .AddressesAndNames
```

- ☞ Setelah daftar *Bluetooth* di sekitar tertangkap kemudian kita pilih maka program berikutnya sebagai berikut :
 - Pilih *listpicker* dan cari *subblock* program *afterpicking* kemudian klik dan pasang pada *viewer*.

```
when ListPicker1 .AfterPicking
do
```

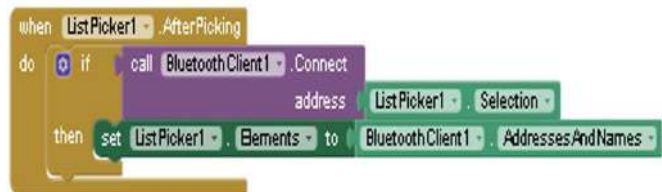


- Jika setelah *Bluetooth* dipilih maka akan terjadi *pairing* antar *Bluetooth*



Blok *if...then* diambil dari *block control*, kemudian cari *Bluetooth connect address* dengan mengklik *Bluetoothclient* kemudian pilih *block connect address*, setelah itu dipasang *list Bluetooth* yang telah dipilih dari block program sebelumnya dengan mengklik *listpicker.selection*.

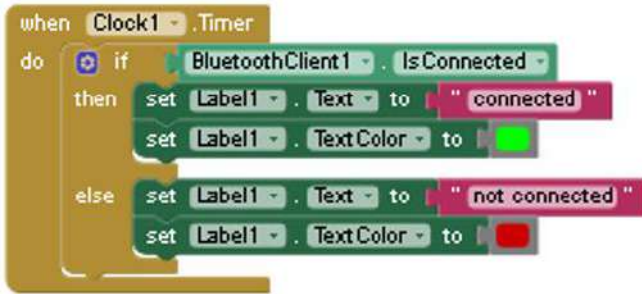
- Kemudian *Bluetooth* yang dipilih dijadikan sebagai elemen dari komponen *listpicker*.



Klik komponen *listpicker* kemudian cari *block* program *listpicker.element*, *drag and drop* ke blok *then* selanjutnya pasang blok nama dan alamat dari *Bluetooth* dengan mengklik komponen *bluetooth* kemudian cari *block bluetoothclient.AddressAndName*, pasang pada elemen *listpicker*.

- ☞ Koneksi antar *Bluetooth* memerlukan waktu, jika tersambung maka akan ditampilkan status “connected” pada label1 dengan warna text hijau, namun jika tidak tersambung antara kedua *Bluetooth* atau gagal koneksi maka pada label1 akan ditampilkan status “not connected” dengan warna merah.



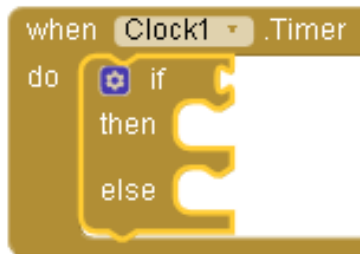


Langkah-langkah membuat blok programnya sebagai berikut :

- Klik komponen clock kemudian pilih block timer dan tempatkan pada viewer.



- Untuk memilih status “connected” atau “not connected” pilih blok control dan pilih blok if...then...else kemudian pasang pada blok clock1.timer.



- Untuk menyatakan kondisi *Bluetooth* terhubung atau tidak dengan perintah blok yang diambil dari komponen *Bluetooth* dan cari blok ***bluetoothclient.isconnected*** dan pasang pada *if* di *blok control*.

- o Jika *Bluetooth* terkoneksi maka status akan dituliskan pada *label1*, cari blok **set label1.text to** kemudian pasang di blok control *then* sebagai berikut,

- o Kemudian untuk menuliskan status diambil blok text " " dengan posisi teks masih kosong kemudian ditulis status "connected".

- o Status terhubung akan diberi warna hijau, pilih komponen *label1* dan cari blok **set label1.textcolor. to** dan pasang di blok perintah *then*.



- o Untuk memberi warna status ambil blok colors kemudian pilih warna hijau.

```

when Clock1 .Timer
do
  if BluetoothClient1 . IsConnected
  then
    set Label1 . Text to "connected"
    set Label1 . TextColor to [Green]
  else

```

- o Untuk perintah blok *else* langkahnya seperti pada perintah blok *then* dengan merubah status *label1* dengan kata “*not connected*” dan diberi warna merah.

```

set Label1 . Text to "not connected"
set Label1 . TextColor to [Red]

```

- o Kemudian pasangkan pada perintah blok *else* sehingga menjadi blok program seperti dibawah ini

```

when Clock1 .Timer
do
  if BluetoothClient1 . IsConnected
  then
    set Label1 . Text to "connected"
    set Label1 . TextColor to [Green]
  else
    set Label1 . Text to "not connected"
    set Label1 . TextColor to [Red]

```

- ☞ Untuk mengendalikan robot dengan perintah tombol-tombol didesain sebanyak lima buah tombol, tombol maju, tombol mundur, tombol belok kanan, tombol belok kiri, dan tombol stop.



```

when Button1 .Click
do
  call BluetoothClient1 .SendText
  text " forward "
  set Label2 . Text to " Maju "
  set robot_atas . Visible to true
  set robot_bawah . Visible to false
  set robot_kiri . Visible to false
  set Robot_kanan . Visible to false
  
```

Langkah-langkah pembuatan tombol hamper semua sama kecuali tombol stop yang tidak menyertakan ilustrasi gambar robot.

- o Pertama klik komponen *button* kemudian pilih blok ***button1.click*** dan pasang pada *viewer*.

```

when Button1 .Click
do
  
```

- o Kemudian untuk mengirimkan teks lewat *Bluetooth*, klik komponen *Bluetooth* dan cari blok ***bluetoothclient.sendtext***, selanjutnya klik blok *text* dari cari blok yang hanya berisi teks saja kemudian pasangkan pada blok *bluetoothclient*.

```

when Button1 .Click
do
  call BluetoothClient1 .SendText
  text " forward "
  
```

- o Untuk menampilkan informasi teks pergerakan robot, maju, mundur, kanan, kiri, atau berhenti menggunakan label2, cari blok ***set label2.text to*** kemudian diisi dengan blok *text*.



```

when Button1 .Click
do
  call BluetoothClient1 .SendText
  text " forward "
  set Label2 . Text to " Maju "

```

- o Untuk memberi tampilan yang interaktif diberikan gambar robot yang diatur sesuai dengan tombol yang ditekan.

```

set robot_atas . Visible to true
set robot_bawah . Visible to false
set robot_kiri . Visible to false
set Robot_kanan . Visible to false

```

Jika robot maju maka diilustrasikan gambar robot mengarah ke atas, berikut ini ilustrasi gambar robot yang digunakan untuk memberikan informasi yang lebih menarik.



- ☞ Langkah untuk tombol dua sama seperti tombol satu, hanya merubah status pada teks saja yaitu menjadi arah kiri atau "*left*".

```

when Button2 .Click
do
  call BluetoothClient1 .SendText
  text " left "
  set Label2 . Text to " Kiri "
  set robot_atas . Visible to false
  set robot_bawah . Visible to false
  set robot_kiri . Visible to true
  set Robot_kanan . Visible to false

```

- ☞ Langkah untuk tombol tiga sama seperti tombol satu, hanya merubah status pada teks saja yaitu menjadi arah kanan atau “right”.

```
when Button3 .Click
do
  call BluetoothClient1 .SendText
  text "right"
  set Label2 .Text to "Kanan"
  set robot_atas .Visible to false
  set robot_bawah .Visible to false
  set robot_kiri .Visible to false
  set Robot_kanan .Visible to true
```

- ☞ Langkah untuk tombol empat sama seperti tombol satu, hanya merubah status pada teks saja yaitu menjadi arah mundur atau “reverse”.


```
when Button4 .Click
do
  call BluetoothClient1 .SendText
  text "reverse"
  set Label2 .Text to "Mundur"
  set robot_atas .Visible to false
  set robot_bawah .Visible to true
  set robot_kiri .Visible to false
  set Robot_kanan .Visible to false
```

- ☞ Untuk tombol 5 tanpa perintah merubah posisi gambar robot hanya menampilkan tulisan berhenti atau “stop” dengan tetap posisi gambar robot yang terakhir.

```
when Button5 .Click
do
  call BluetoothClient1 .SendText
  text "stop"
  set Label2 .Text to "Berhenti"
```

- ☞ Selain perintah menggunakan tombol, robot ini juga dapat diperintah dengan menggunakan suara atau *voice recognition* dengan menekan tombol kemudian mengucapkan kata yang



digunakan sebagai perintah untuk menggerakkan robot. Tombol perintah voice sebagai berikut. 

- Langkah-langkah membuat blok program untuk voice recognition sebagai berikut :
 - Pilih komponen tombol yaitu **button6** kemudian cari blok program *button.click* taruh di *viewer*.

```
when Button6 .Click
do
```

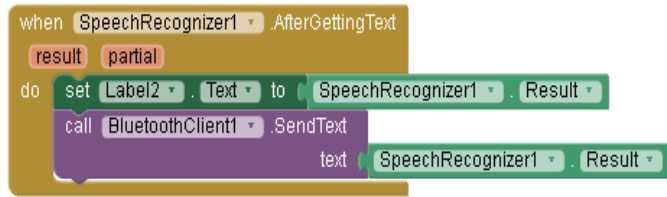
- Selanjutnya cari blok program untuk *speechrecognizer* dengan mengklik komponen **speechrecognizer1** lalu cari blok berwarna ungu **call *speechrecognizer.getText*** yang berfungsi untuk mengubah suara menjadi teks.

```
when Button6 .Click
do call SpeechRecognizer1 .GetText
```

- Sebelum prosedur *speechrecognizer* mengambil teks maka status pada label2 dikosongkan dahulu, blok program seperti dibawah ini.

```
when SpeechRecognizer1 .BeforeGettingText
do set Label2 .Text to ""
```

- Setelah mengambil text kemudian prosedur *speechrecognizer* akan memberikan status kepada label2 sesuai yang diucapkan dan juga mengirimkan ke bluetooth untuk memerintahkan robot.



Berikut ini program lengkap arduino dan aplikasi control mobile robot:

```
#include <SoftwareSerial.h>
SoftwareSerial BT (8, 7); //TX, RX
String readdata;

int IA1 = 5;
int IA2 = 6;
int ENA = 9;

int IB1 = 3;
int IB2 = 4;
int ENB = 2;

void setup() {
    BT.begin(9600);

    // put your setup code here, to run once:
    pinMode(IA1, OUTPUT);
    pinMode(IA2, OUTPUT);
    pinMode(ENA, OUTPUT);
    pinMode(IB1, OUTPUT);
    pinMode(IB2, OUTPUT);
    pinMode(ENB, OUTPUT);
}

void loop() {

    while (BT.available()) { //Check if there is an
```



available byte to read

```

    delay(10); //Delay added to make thing stable
    char c = BT.read(); //Conduct a serial read
    readdata += c; // build the string- "forward",
    "reverse", "left" and "right"
}

```

```

if (readdata.length() > 0) {

    // if data received as forward move robot forward
    if (readdata == "maju")
    {
        Robot_Maju();
        delay(100);
    }
// if data received as reverse move robot reverse
    else if (readdata == "mundur")
    {
        Robot_Mundur();
        delay(100);
    }
// if data received as right turn robot to right direction.
    else if (readdata == "kanan")
    {
        Robot_Belok_Kanan();
        delay(100);
    }
// if data received as left turn robot to left direction
    else if (readdata == "kiri")
    {
        Robot_Belok_Kiri();
        delay(100);
    }
}

```



```

}
// if data received as stop, halt the robot
else if (readdata == "stop")
{
    Robot_Berhenti();
    delay(100);
}
readdata="";}} // Reset the variable
// code ends here

```

```

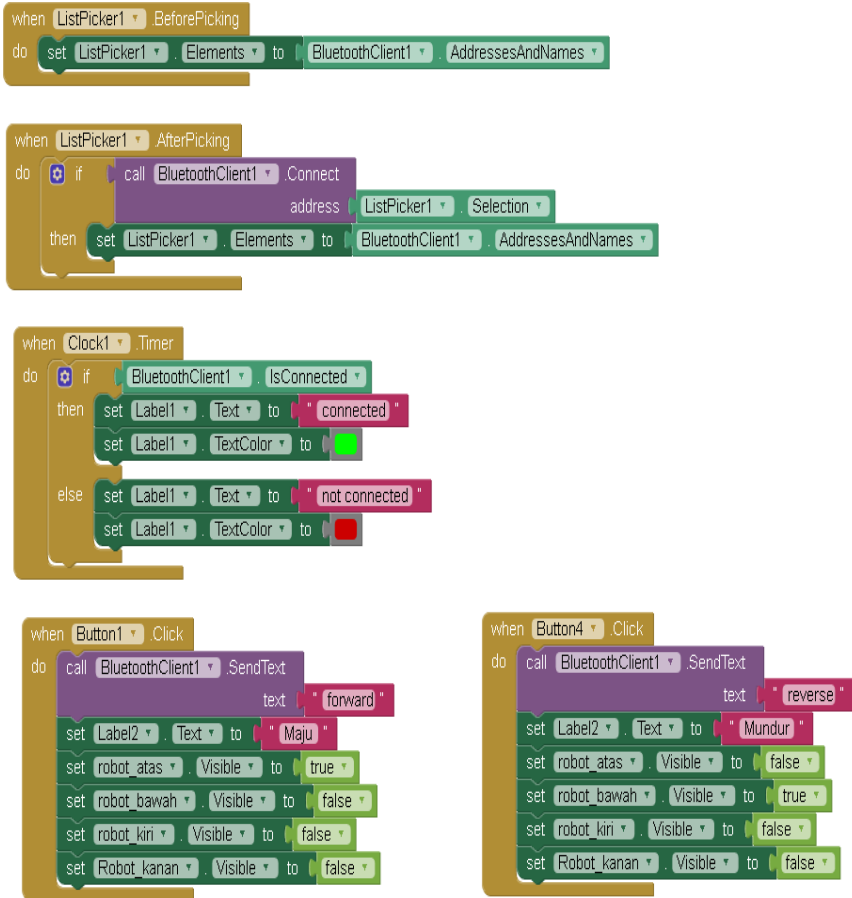
void Robot_Mundur()
{
    digitalWrite (IB1, HIGH);
    digitalWrite (IB2, LOW);
    analogWrite (ENB, 150);
    digitalWrite (IA1, HIGH);
    digitalWrite (IA2, LOW);
    analogWrite (ENA, 150);
}
void Robot_Belok_Kanan()
{
    digitalWrite (IB1, LOW);
    digitalWrite (IB2, HIGH);
    analogWrite (ENB, 150);
    digitalWrite (IA1, HIGH);
    digitalWrite (IA2, LOW);
    analogWrite (ENA, 0);
}
void Robot_Belok_Kiri()
{
    digitalWrite (IB1, HIGH);
    digitalWrite (IB2, LOW);
    analogWrite (ENB, 0);
}

```



```
    digitalWrite(IA1, LOW);  
    digitalWrite(IA2, HIGH);  
    analogWrite(ENA, 150);  
}  
void Robot_Maju()  
{  
    digitalWrite(IB1, LOW);  
    digitalWrite(IB2, HIGH);  
    analogWrite(ENB, 150);  
    digitalWrite(IA1, LOW);  
    digitalWrite(IA2, HIGH);  
    analogWrite(ENA, 150);  
}  
  
void Robot_Berhenti()  
{  
    digitalWrite(IB1, LOW);  
    digitalWrite(IB2, HIGH);  
    analogWrite(ENB, 0);  
    digitalWrite(IA1, LOW);  
    digitalWrite(IA2, HIGH);  
    analogWrite(ENA, 0);  
}
```





The image displays four Scratch code blocks for controlling a mobile robot. The first block, 'when ListPicker1 BeforePicking', sets the elements of ListPicker1 to BluetoothClient1's addresses and names. The second block, 'when ListPicker1 AfterPicking', checks if BluetoothClient1 is connected; if yes, it connects and updates the list; if no, it sets the text to 'not connected'. The third block, 'when Clock1 Timer', checks the connection status and updates Label1's text and color (green for connected, red for not connected). The fourth block, 'when Button1 Click', sends 'forward' to BluetoothClient1 and sets robot visibility (robot_atas true, others false). The fifth block, 'when Button4 Click', sends 'reverse' to BluetoothClient1 and sets robot visibility (robot_bawah true, others false).

```
when ListPicker1 BeforePicking
do
  set ListPicker1 Elements to BluetoothClient1 AddressesAndNames

when ListPicker1 AfterPicking
do
  if call BluetoothClient1 Connect
    address ListPicker1 Selection
  then
    set ListPicker1 Elements to BluetoothClient1 AddressesAndNames

when Clock1 Timer
do
  if BluetoothClient1 IsConnected
  then
    set Label1 Text to "connected"
    set Label1 TextColor to green
  else
    set Label1 Text to "not connected"
    set Label1 TextColor to red

when Button1 Click
do
  call BluetoothClient1 SendText
  text "forward"
  set Label2 Text to "Maju"
  set robot_atas Visible to true
  set robot_bawah Visible to false
  set robot_kiri Visible to false
  set Robot_kanan Visible to false

when Button4 Click
do
  call BluetoothClient1 SendText
  text "reverse"
  set Label2 Text to "Mundur"
  set robot_atas Visible to false
  set robot_bawah Visible to true
  set robot_kiri Visible to false
  set Robot_kanan Visible to false
```

Gambar 30. Program Lengkap Kontrol Mobile Robot



```

when Button2 Click
do
  call BluetoothClient1 SendText
    text left
  set Label2 Text to Kiril
  set robot_atas Visible to false
  set robot_bawah Visible to false
  set robot_kiri Visible to true
  set Robot_kanan Visible to false

when Button3 Click
do
  call BluetoothClient1 SendText
    text right
  set Label2 Text to Kanan
  set robot_atas Visible to false
  set robot_bawah Visible to false
  set robot_kiri Visible to false
  set Robot_kanan Visible to true

when Button5 Click
do
  call BluetoothClient1 SendText
    text stop
  set Label2 Text to Berhenti

when Button6 Click
do
  call SpeechRecognizer1 GetText

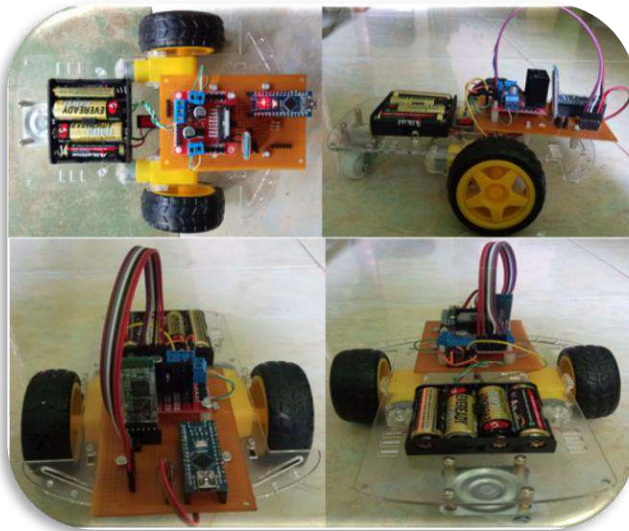
when SpeechRecognizer1 BeforeGettingText
do
  set Label2 Text to

when SpeechRecognizer1 AfterGettingText
result partial
do
  set Label2 Text to SpeechRecognizer1 Result
  call BluetoothClient1 SendText
    text SpeechRecognizer1 Result

```

Gambar 30. Program Lengkap Kontrol Mobile Robot (lanjutan)

Hasil desain *mobile robot* serta aplikasi kontrol menggunakan *smartphone* adalah sebagai berikut :

Gambar 31. *Mobile Robot*



Gambar 32. *Smartphone Kontrol Mobile Robot*



DAFTAR PUSTAKA

- Dodit Suprianto & Rini Agustina, S.Kom, M.Pd** , 2012, *Pemrograman Aplikasi Android*, MediaKom, ISBN – 9789798772795
- Edward Mitchell**, 2016, *App Inventor 2 Graphics, Animation and Charts*, Kindle Direct Publishing, ISBN –9781518614279
- Endra Pitowarno**, 2006, *Robotika : Desain, Kontrol, Dan Kecerdasan Buatan*, Penerbit Andi, Yogyakarta, ISBN – 979-763-094-3
- Eueung Mulyana**, 2012, *App Inventor: Ciptakan Sendiri Aplikasi Androidmu*, Andi Publisher, ISBN –9789792931686
- Hari Santoso, 2015, Panduan Praktis Arduino untuk Pemula, ebook www.elangsakti.com
- Jason Torchinsky, Beau Boeckmann**, 2019, *Robot, Take the Wheel: The Road to Autonomous Cars and the Lost Art of Driving*, Appollo Publisher, USA, ISBN – 978-1-948062-27-5
- Mohammad Eskatul Islam**, 2011 , *Design and Fabrication of a Line Follower Robot*, LAP Lambert Academic Publishing, ISBN-13: 978-3845407647
- Nazruddin Safaat H., 2011, Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android, Penerbit Informatika, Bandung, ISBN – 978-602-8758-23-9
- Saludin Muis**, 2011, *Prinsip Dasar Cara Kerja Robot*, Graha Ilmu, Yogyakarta, ISBN : 978-979-756-724-8
- Widodo Budhiharto**, 2007, *Belajar Sendiri Membuat Robot Cerdas*, Elex Media Komputindo, Jakarta, ISBN – 979-20-9178-5
- <https://appinventor.mit.edu>
www.arduino.cc



DAFTAR PUSTAKA

<https://www.elangsakti.com/2017/11/belajar-arduino.html>
<https://repository.unej.ac.id/handle/123456789/84859>



TENTANG PENULIS



Slamet Winardi, ST, MT terlahir di kota Semarang 3 Agustus 1971, mengenyam pendidikan terakhir di Sistem Kontrol ITS lulus tahun 2003. Sekarang menekuni bidang Internet of Things khususnya IoT transportasi. Prestasi terakhir yang dihasilkan sebuah paten , 2 merek, lebih dari 21 hak cipta, dan 9 buku dari sebuah penelitian yang didanai oleh Ristek Dikti. Konsen dibidangnya dengan membentuk komunitas IoT yang diberi nama Indonesian Internet of Things yang disingkat Id-iot.



Dr. Ir. Kunto eko susilo, MT, lahir di Jember 3 pebruari 1969. Pendidikan Magister di bidang sistem kontrol dari ITS lulus tahun 1998, dan program doktor ilmu ekonomi untag surabaya lulus tahun 2018. Peminatan penelitian di bidang instrumentasi dan kontrol selain aktif di asosiasi profesi bidang mekanikal dan elektrik sebagai ahli madya dan asesor.



Arief Budijanto, ST.,MT. dilahirkan di kota Surabaya 23 April 1969, mengenyam pendidikan terakhir di Teknik Komputer Institut Teknologi Bandung lulus tahun 2007. Sekarang menekuni bidang Embedded System, Real-Time System. Prestasi terakhir memdapatkan penelitian hibah bersaing pada bidang Embedded Digital Control yang di danai oleh DIKITI. Selain itu sebagai dosen pada perguruan tinggi Politeknik NSC, Univ. Widya Kartika, Univ Narotama dan Politeknik Pelayaran Surabaya.



TENTANG PENULIS



Tresna Maulana Fahrudin, S.ST., M.T., Lahir di Gresik, 1 Mei 1993, menyelesaikan jenjang Magister Terapan di bidang Knowledge Discovery dan Intelligent Computing, Politeknik Elektronika Negeri Surabaya tahun 2017. Beberapa penelitiannya sudah dipublikasikan di beberapa Seminar Nasional dan Internasional hingga mendapatkan Best Paper Award pada kategori Intelligent Computing for Medical Engineering and Analysis di KCIC 2016. Selain itu, berkesempatan mengikuti Digital Talent Scholarship Training 2019 yang diselenggarakan oleh Kominfo dan Microsoft dengan mengambil skema Data Science dan Data Analytic. Saat ini menekuni bidang penelitian Machine Learning dan Deep Learning yang diaplikasikan ke berbagai bentuk dataset.

