

BAB II

LANDASAN TEORI

2.1 Robotika

Istilah robot berasal dari bahasa Ceko Slowakia. Kata robot berasal dari kosakata “Robota” yang berarti “kerja cepat”. Istilah ini muncul pada tahun 1920 oleh seorang pengarang sandiwara bernama Karel Capek. Karyanya pada saat itu berjudul “Rossum’s Universal Robot” yang artinya Robot Dunia milik Rossum. Rossum merancang dan membangun suatu bala tentara yang terdiri dari robot industri yang akhirnya menjadi terlalu cerdas dan akhirnya menguasai manusia.

Kata Robotika juga berasal dari novel fiksi sains “runaround” yang ditulis oleh Isaac Asimov pada tahun 1942. Sedangkan pengertian robot secara tepat adalah sistem atau alat yang dapat berperilaku atau meniru perilaku manusia dengan tujuan untuk menggantikan dan mempermudah kerja/aktifitas manusia. Untuk dapat diklasifikasikan sebagai robot, maka robot harus memiliki dua macam kemampuan yaitu:

- 1) Bisa mendapatkan informasi dari sekelilingnya.
- 2) Bisa melakukan sesuatu secara fisik seperti bergerak atau memanipulasi objek.

Untuk dapat dikatakan sebagai robot sebuah sistem tidak perlu untuk meniru semua tingkah laku manusia, namun suatu sistem tersebut dapat mengadopsi satu atau dua dari sistem yang ada pada diri manusia saja sudah dapat dikatakan sebagai robot. Sistem yang diadopsi dapat berupa sistem penglihatan (mata), sistem pendengaran (telinga) ataupun sistem gerak. Sebuah robot dapat saja dibuat

untuk berbagai macam aktifitas, namun sebuah robot harus dibuat dengan tujuan untuk kebaikan manusia.

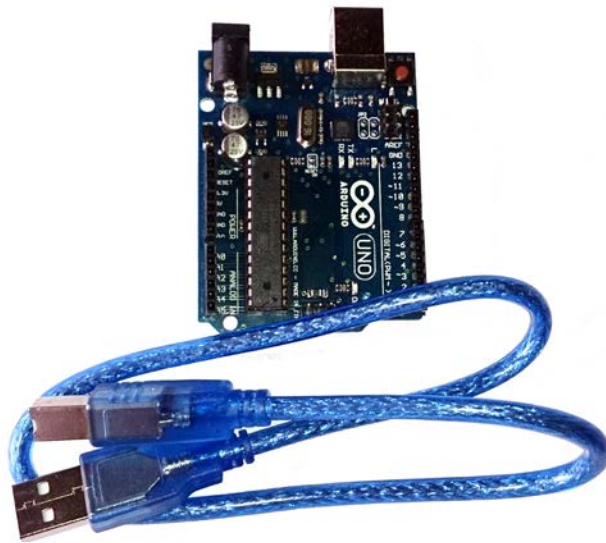
Ada beberapa fungsi robot, sehingga manusia memerlukan kehadirannya yaitu:

1. Meningkatkan produksi, akurasi dan daya tahan. Robot ini banyak digunakan di industri.
2. Untuk tugas-tugas yang berbahaya, kotor dan beresiko. Robot ini digunakan ketika manusia tidak mampu masuk ke daerah yang beresiko. Seperti Robot Untuk menjelajah planet, robot untuk mendeteksi limbah nuklir, robot militer dll.
3. Untuk pendidikan. Banyak robot yang digunakan untuk menarik pelajar belajar teknologi seperti robot lego, dll.

Untuk menolong manusia. Seperti di rumah untuk membersihkan rumah pakai penghisap debu otomatis, di rumah sakit untuk menghantar makanan, membantu operasi, dll.

2.2 Mikrokontroler

Mikrokontroler merupakan suatu *device* yang didalamnya sudah terintegrasi dengan *I/O Port*, RAM, ROM, sehingga dapat digunakan untuk berbagai keperluan kontrol. Secara internal mikrokontroler terdiri atas unit – unit fungsional *arithmetic and logical unit*. Ada beberapa tipe mikrokontroler antara lain ATmega8535, ATmega16, ATmega32, ATmega328, yang membedakan antara mikrokontroler antara lain adalah, ukuran memori, banyaknya GPIO (pin *input/output*), peripheral (USART, *timer*, *counter*, dll).



Gambar 2.1 Mikrokontroler Arduino Uno R3 ATmega328P

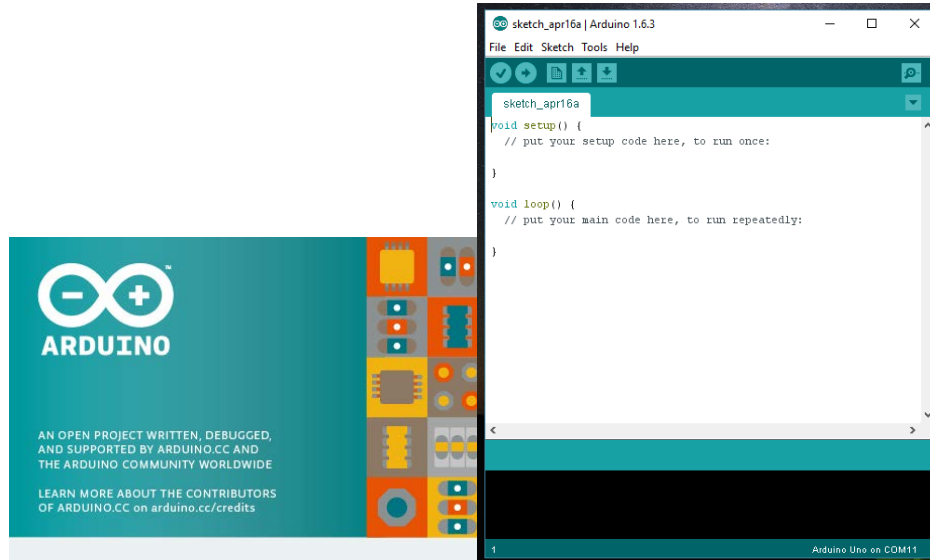
Berikut adalah spesifikasi umum dari mikrokontroler :

- 1) Mikrokontroler : ATMEGA328
- 2) Tegangan Operasi : 5V
 - a) Tegangan *Input (recommended)* : 7 - 12 V
 - b) Tegangan *Input (limit)* : 6 - 20 V
- 3) Pin *digital I/O* : 14 (6 diantaranya pin PWM)
- 4) Pin Analog *input* : 6
- 5) Arus DC per pin I/O : 20 mA
- 6) Arus DC untuk pin 3.3 V : 50 mA
- 7) *Flash Memory* : 32 KB (Atmega 328P) dengan 0.5KB digunakan untuk *bootloader*
- 8) SRAM : 2 KB (Atmega 328P)
- 9) EEPROM : 1 KB (Atmega 328P)
- 10) Kecepatan Pewaktuan : 16 Mhz

- 11) Memiliki koneksi USB (menggunakan ATmega8U2 sebagai konverter USB to Serial)
- 12) Antarmuka : UART TTL, I2C, SPI dan USB (Virtual Com)
- 13) Pemrograman menggunakan Arduino *Software* (berbasis bahasa C yang telah dilengkapi dengan *library* yang kompatibel dengan desain *hardware* Arduino)
- 14) Pengisian kode program dapat menggunakan koneksi USB

2.3 Perangkat lunak

Software merupakan perangkat lunak yang tidak dapat disentuh maupun dilihat wujud fisiknya. Sekalipun tidak berwujud fisik, keberadaan software sangat penting. Istilah software sebenarnya mengacu ke sekumpulan instruksi dan data komputer yang terorganisasi. Software dibuat menggunakan bahasa pemrograman dan utilitas terkait. Bahasa pemrograman dan utilitas terkait ini bisa dalam bentuk program tunggal seperti script interpreter, dalam bentuk paket yang di dalamnya terdapat compiler, linker dan tool pendukung lainnya atau dalam bentuk IDE (Integrated Development Environment) seperti NetBean yang di dalamnya sudah terdapat editor, debugger dan tool pendukung untuk banyak bahasa pemrograman. Software dapat dibagi berdasarkan jenis dan pendistribusiannya.



Gambar 2.2 Arduino Software (IDE)

2.3.1 IDE Arduino

IDE Arduino adalah software yang sangat canggih ditulis dengan menggunakan Java. IDE Arduino terdiri dari:

- Editor program, sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa Processing.
- Compiler, sebuah modul yang mengubah kode program (bahasa Processing) menjadi kode biner. Bagaimanapun sebuah microcontroller tidak akan bisa memahami bahasa Processing. Yang bisa dipahami oleh microcontroller adalah kode biner. Itulah sebabnya compiler diperlukan dalam hal ini.
- Uploader, sebuah modul yang memuat kode biner dari Jomputer ke dalam memory di dalam papan Arduino.

Arduino ini bisa dijalankan di komputer dengan berbagai macam *platform* karena didukung atau berbasis Java. *Source* program yang dibuat untuk aplikasi

mikrokontroler adalah bahasa C/C++ dan dapat digabungkan dengan assembly.

(arduino.cc, 2013)

Structure	Variables	Functions
<ul style="list-style-type: none"> + <code>setup()</code> + <code>loop()</code> <p>Control Structures</p> <ul style="list-style-type: none"> + <code>if</code> + <code>if...else</code> + <code>for</code> + <code>switch case</code> + <code>while</code> + <code>do... while</code> + <code>break</code> + <code>continue</code> + <code>return</code> + <code>goto</code> 	<p>Constants</p> <ul style="list-style-type: none"> + <code>HIGH LOW</code> + <code>INPUT OUTPUT</code> + <code>true false</code> + <code>integer constants</code> + <code>floating point constants</code> <p>Data Types</p> <ul style="list-style-type: none"> + <code>void</code> + <code>boolean</code> + <code>char</code> + <code>unsigned char</code> + <code>byte</code> + <code>int</code> 	<p>Digital I/O</p> <ul style="list-style-type: none"> + <code>pinMode()</code> + <code>digitalWrite()</code> + <code>digitalRead()</code> <p>Analog I/O</p> <ul style="list-style-type: none"> + <code>analogReference()</code> + <code>analogRead()</code> + <code>analogWrite() - PWM</code> <p>Advanced I/O</p> <ul style="list-style-type: none"> + <code>tone()</code> + <code>noTone()</code> + <code>shiftOut()</code> + <code>shiftIn()</code>

Gambar 2.3 Bahasa Pemrogramana Arduino

Struktur

Setiap program Arduino (biasa disebut *sketch*) mempunyai dua buah fungsi yang harus ada.

- **`void setup() { }`**
 - Semua kode didalam kurung kurawal akan dijalankan hanya satu kali ketika program Arduino dijalankan untuk pertama kalinya.
- **`void loop() { }`**
 - Fungsi ini akan dijalankan setelah setup (fungsi *void setup*) selesai. Setelah dijalankan satu kali fungsi ini akan dijalankan lagi, dan lagi secara terus menerus sampai catu daya (*power*) dilepaskan.

Syntax

Berikut ini adalah elemen bahasa C yang dibutuhkan untuk format penulisan.

- `//(komentar satu baris)`

- Kadang diperlukan untuk memberi catatan pada diri sendiri apa arti dari kode-kode yang dituliskan. Cukup menuliskan dua buah garis miring dan apapun yang kita ketikkan dibelakangnya akan diabaikan oleh program.
- `/* */`(komentar banyak baris)
 - Jika anda punya banyak catatan, maka hal itu dapat dituliskan pada beberapa baris sebagai komentar. Semua hal yang terletak di antara dua simbol tersebut akan diabaikan oleh program.
- `{ }`(kurung kurawal)
 - Digunakan untuk mendefinisikan kapan blok program mulai dan berakhir (digunakan juga pada fungsi dan pengulangan).
- `;`(titik koma)
 - Setiap baris kode harus diakhiri dengan tanda titik koma (jika ada titik koma yang hilang maka program tidak akan bisa dijalankan).

Variabel

Sebuah program secara garis besar dapat didefinisikan sebagai instruksi untuk memindahkan angka dengan cara yang cerdas. Variabel inilah yang digunakan untuk memindahkannya.

- **int** (integer)
 - Digunakan untuk menyimpan angka dalam 2 byte (16 bit). Tidak mempunyai angka desimal dan menyimpan nilai dari -32,768 dan 32,767.
- **long** (long)

- Digunakan ketika integer tidak mencukupi lagi. Memakai 4 byte (32 bit) dari memori (RAM) dan mempunyai rentang dari -2,147,483,648 dan 2,147,483,647.
- **boolean** (boolean)
 - Variabel sederhana yang digunakan untuk menyimpan nilai *TRUE* (benar) atau *FALSE* (salah). Sangat berguna karena hanya menggunakan 1 bit dari RAM.
- **float** (float)
 - Digunakan untuk angka desimal (floating point). Memakai 4 byte (32 bit) dari RAM dan mempunyai rentang dari -3.4028235E+38 dan 3.4028235E+38.
- **char** (character)
 - Menyimpan 1 karakter menggunakan kode ASCII (misalnya 'A' = 65). Hanya memakai 1 byte (8 bit) dari RAM.

Operator Matematika

Operator yang digunakan untuk memanipulasi angka (bekerja seperti matematika yang sederhana).

- =
 - Membuat sesuatu menjadi sama dengan nilai yang lain (misalnya: $x = 10 * 2$, x sekarang sama dengan 20).
- %
 - Menghasilkan sisa dari hasil pembagian suatu angka dengan angka yang lain (misalnya: $12 \% 10$, ini akan menghasilkan angka 2).
- +

- Penjumlahan
- -
- Pengurangan
- *
- Perkalian
- /
- Pembagian

Operator Pembanding

Digunakan untuk membandingkan nilai logika.

- ==
 - Sama dengan (misalnya: $12 == 10$ adalah FALSE (salah) atau $12 == 12$ adalah TRUE (benar))
- !=
 - Tidak sama dengan (misalnya: $12 != 10$ adalah TRUE (benar) atau $12 != 12$ adalah FALSE (salah))
- <
 - Lebih kecil dari (misalnya: $12 < 10$ adalah FALSE (salah) atau $12 < 12$ adalah FALSE (salah) atau $12 < 14$ adalah TRUE (benar))
- >
 - Lebih besar dari (misalnya: $12 > 10$ adalah TRUE (benar) atau $12 > 12$ adalah FALSE (salah) atau $12 > 14$ adalah FALSE (salah))

Struktur Pengaturan

Program sangat tergantung pada pengaturan apa yang akan dijalankan berikutnya, berikut ini adalah elemen dasar pengaturan (banyak lagi yang lain dan bisa dicari di internet).

1. **if..else**, dengan format seperti berikut ini:

```
if (kondisi) { }
```

```
else if (kondisi) { }
```

```
else { }
```

Dengan struktur seperti diatas program akan menjalankan kode yang ada di dalam kurung kurawal jika kondisinya TRUE, dan jika tidak (FALSE) maka akan diperiksa apakah kondisi pada *else if* dan jika kondisinya FALSE maka kode pada *else* yang akan dijalankan.

2. **for**, dengan format seperti berikut ini:

```
for (int i = 0; i < #pengulangan; i++) { }
```

Digunakan ingin melakukan pengulangan kode di dalam kurung kurawal beberapa kali, ganti #pengulangan dengan jumlah pengulangan yang diinginkan. Melakukan penghitungan ke atas dengan *i++* atau ke bawah dengan *i--*.

Digital

1. **pinMode(pin, mode)**

Digunakan untuk menetapkan mode dari suatu pin, *pin* adalah nomor pin yang akan digunakan dari 0-19 (pin analog 0-5 adalah 14-19). Mode yang bisa digunakan adalah *INPUT* atau *OUTPUT*.

2. **digitalWrite(pin, value)**

Ketika sebuah pin ditetapkan sebagai *OUTPUT*, pin tersebut dapat dijadikan *HIGH* (ditarik menjadi 5 volts) atau *LOW* (diturunkan menjadi ground).

3. **digitalRead(pin)**

Ketika sebuah pin ditetapkan sebagai *INPUT* maka anda dapat menggunakan kode ini untuk mendapatkan nilai pin tersebut apakah *HIGH* (ditarik menjadi 5 volts) atau *LOW* (diturunkan menjadi ground).

Analog

Arduino adalah mesin digital tetapi mempunyai kemampuan untuk beroperasi di dalam alam analog (mengggunakan trik). Berikut ini cara untuk menghadapi hal yang bukan digital.

1. **analogWrite(pin, value)**

Beberapa pin pada Arduino mendukung PWM (pulse width modulation) yaitu pin 3, 5, 6, 9, 10, 11. Ini dapat merubah pin hidup (*on*) atau mati (*off*) dengan sangat cepat sehingga membuatnya dapat berfungsi layaknya keluaran analog. *Value* (nilai) pada format kode tersebut adalah angka antara 0 (0% duty cycle ~ 0V) dan 255 (100% duty cycle ~ 5V).

2. **analogRead(pin)**

Ketika pin analog ditetapkan sebagai *INPUT* anda dapat membaca keluaran voltase-nya. Keluarannya berupa angka antara 0 (untuk 0 volts) dan 1024 (untuk 5 volts).

2.3.2 **Android Studio**

Android Studio adalah Lingkungan Pengembangan Terpadu - Integrated Development Environment (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA . Selain merupakan editor kode IntelliJ dan alat

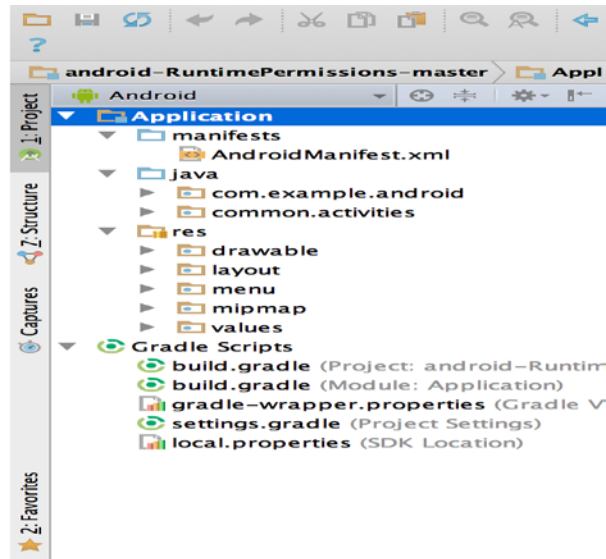
pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas saat membuat aplikasi Android, misalnya:

- Sistem pembuatan berbasis Gradle yang fleksibel
- Emulator yang cepat dan kaya fitur
- Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
- Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
- Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
- Alat penguji dan kerangka kerja yang ekstensif
- Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
- Dukungan C++ dan NDK
- Dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud Messaging dan App Engine

Struktur Proyek

Setiap proyek di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file sumber daya. Jenis-jenis modul mencakup:

- Modul aplikasi Android
- Modul perpustakaan
- Modul Google App Engine



Gambar 2.4 File Proyek Di Tampilan Android.

Secara default, Android Studio menampilkan file proyek dalam tampilan proyek Android seperti yang ditunjukkan dalam gambar 2.4 Tampilan ini diatur menurut modul untuk memberi akses cepat ke file sumber kunci proyek.

Semua file versi terlihat di bagian atas di bawah **Gradle Scripts** dan masing-masing modul aplikasi berisi folder berikut:

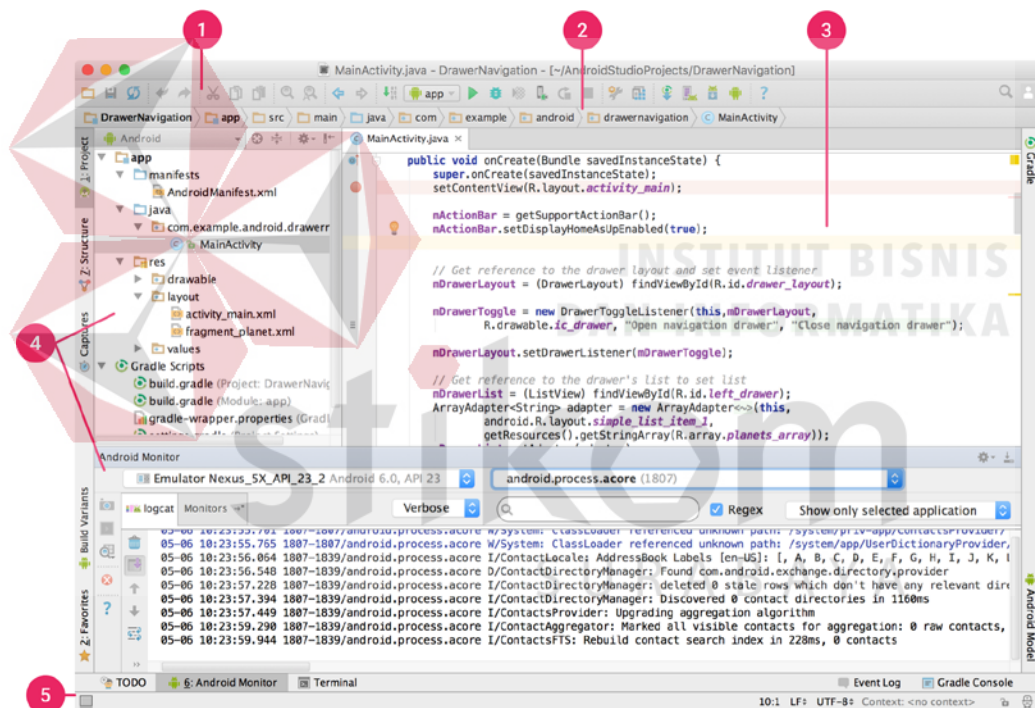
- **manifests:** Berisi file AndroidManifest.xml.
- **java:** Berisi file kode sumber Java, termasuk kode pengujian JUnit.
- **res:** Berisi semua sumber daya bukan kode, seperti tata letak XML, string UI, dan gambar bitmap.

Struktur proyek Android pada disk berbeda dari representasi rata ini. Untuk melihat struktur file sebenarnya dari proyek ini, pilih Project dari menu tarik turun Project.

Juga bisa menyesuaikan tampilan file proyek untuk berfokus pada aspek tertentu dari pengembangan aplikasi. Misalnya, memilih tampilan Problems dari tampilan proyek akan menampilkan tautan ke file sumber yang berisi kesalahan pengkodean dan sintaks yang dikenal, misalnya tag penutup elemen XML tidak ada dalam file tata letak.

Antarmuka Pengguna

Jendela utama Android Studio terdiri dari beberapa bidang logika yang diidentifikasi dalam gambar 2.5



Gambar 2.5 Tampilan Bidang Kerja Pada Android Studio

1. Toolbar memungkinkan untuk melakukan berbagai jenis tindakan, termasuk menjalankan aplikasi dan meluncurkan alat Android.

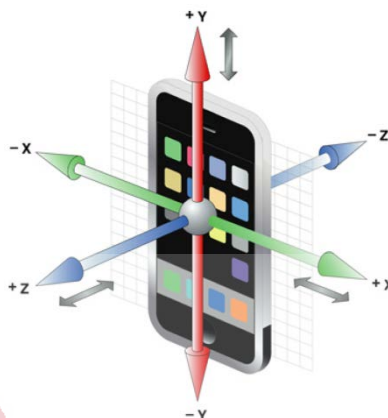
2. Navigasi akan membantu menavigasi di antara proyek dan file yang dibuka untuk pengeditan. Di sini tampilan struktur yang terlihat tampak lebih ringkas daripada jendela Project.
3. Jendela editor merupakan tempat membuat dan mengubah kode. Bergantung pada jenis file saat ini, editor dapat berubah. Misalnya, ketika melihat file tata letak, editor akan menampilkan Layout Editor.
4. Jendela alat memberi akses ke tugas-tugas spesifik seperti pengelolaan proyek, penelusuran, kontrol versi, dan banyak lagi. Bisa meluaskan dan juga menciutkannya.
5. Status menampilkan status proyek Anda dan IDE itu sendiri, serta setiap peringatan atau pesan.

Pada android studio bisa menata jendela utama untuk memberi ruang layar yang lebih luas dengan menyembunyikan atau memindahkan bilah alat dan jendela alat. Juga bisa menggunakan pintasan keyboard untuk mengakses sebagian besar fitur IDE.

Dapat menelusuri seluruh kode sumber, basis data, tindakan, elemen antarmuka pengguna, dan seterusnya setiap saat dengan menekan tombol Shift dua kali, atau mengklik kaca pembesar di sudut kanan atas dari jendela Android Studio. Ini akan sangat berguna misalnya saat mencoba menemukan tindakan IDE tertentu yang lupa cara memicunya.

2.4 Sensor Accelerometer Pada Smartphone

Salah satu sensor yang terdapat pada smartphone ialah sensor accelerometer, yang mana accelerometer sendiri adalah perangkat kompak yang dirancang untuk mengukur percepatan dynamic (objek bergerak) dan static (gravitasi bumi), serta mengukur sudut kemiringan.



Gambar 2.6 Sumbu Koordinat Accelerometer Smartphone

2.4.1 Prinsip Kerja Sensor Accelerometer

Prinsip kerja dari transduser ini berdasarkan hukum fisika bahwa apabila suatu konduktor digerakkan melalui suatu medan magnet, atau jika suatu medan magnet digerakkan melalui suatu konduktor, maka akan timbul suatu tegangan induksi pada konduktor tersebut. Accelerometer akan berinteraksi dengan gravitasi bumi, pada kondisi tegak lurus tersebut accelerometer mengalami percepatan sebesar $1g$. Jika kondisi EM satelit miring, accelerometer akan mengalami percepatan sebesar $1g$ dikalikan dengan $\sin \theta$.

2.4.2 Fungsi Accelerometer Pada Smartphone

1. Accelerometer ponsel ada yang namanya fungsi shake control, dengan fungsi ini maka dengan hanya menggoyangkan ponsel kita bisa

mengubah fitur, misalnya mengubah lagu yang sedang di putar ke lagu selanjutnya atau lagu sebelumnya.

2. Video Games yang menggunakan Wii Remote kontroler yang berisi accelerometer dan dirancang terutama untuk input gerak. Pengguna juga memiliki pilihan untuk membeli sebuah gerak-sensitif tambahan, sehingga masukan gerak bisa dicatat dari kedua tangan pengguna.
3. Untuk kontrol antarmuka pengguna, sering accelerometer digunakan untuk menyajikan pandangan landscape atau potret layar perangkat, berdasarkan cara perangkat yang sedang digunakan. Misalnya mengganti tampilan layar dari vertical (portrait) menjadi horizontal (landscape).
4. Kamera video juga memanfaatkan accelerometer untuk menstabilkan tangkapan gambar (image stabilization), untuk anti-blur saat menangkap gambar.
5. Pada kamera foto memanfaatkannya untuk mencegah hasil pemotretan buram. Contoh penggunaan aplikasi ini adalah seperti yang terdapat pada Handphone Android masa kini. Beberapa kamera digital seperti Canon PowerShot dan Ixus dilengkapi dengan accelerometer untuk mengatur format foto dan merotasi hasil pemotretan saat preview.
6. Penggunaan lain accelerometer adalah sebagai pedometer , atau penghitung langkah. Dipakai di banyak gadget sebagai monitor banyaknya langkah saat berolah raga. Apple dan Nike bekerjasama mengeluarkan produk sepatu yang bisa berkomunikasi dengan pemutar lagu Ipod, mengirimkan banyak langkah, jumlah kalori terbakar dan waktu bergerak.

Kelebihan dan Kekurangan Accelerometer yaitu:

1. Hanya dapat membaca dalam 2 sumbu, yaitu atas dan bawah karena dipengaruhi oleh gravitasi
2. Membutuhkan penggunaan listrik yang tinggi (boros baterai)
3. Gerakan lebih patah-patah/ tidak halus seperti pada gyroscope
4. accelerometer tidak bisa mengikuti pergerakan yang cepat dikarenakan responnya yang lamban.

Accelerometer dapat memberikan pengukuran sudut tegak lurus yang akurat ketika sistem sedang diam (statis).

2.5 Motor DC

Motor *brushless* adalah motor yang mempunyai kekuatan yang konsisten dan kinerja dari *run* untuk menjalankan sekaligus mudah dalam pemeliharaan serta terdapat sistem pengontrol kecepatan listrik. Dalam motor brushless terdapat tiga komponen diantaranya adalah.



Gambar 2.7 Motor DC 12v

1. Kutub medan

Secara sederhana digambarkan bahwa interaksi dua kutub magnet yang menyebabkan perputaran pada motor DC. Motor DC memiliki kutub medan yang stasioner dan dinamo yang menggerakkan bearing pada ruang diantara kutub medan. Motor DC sederhana memiliki dua kutub medan kutub utara dan kutub selatan. Garis magnetic energy membesar melintasi bukaan diantara kutub-kutub dari utara ke selatan. Untuk motor yang lebih besar atau lebih kompleks terdapat satu atau lebih electromagnet. Elektromagnet menerima listrik dari sumber daya dari luar sebagai penyedia struktur medan

2. Dinamo

Bila arus masuk menuju dinamo, maka arus ini akan menjadi elektromagnet. Dinamo yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakkan beban. Untuk kasus motor DC yang kecil, dinamo berputar dalam medan magnet yang dibentuk oleh kutub – kutub, sampai kutub utara dan selatan magnet berganti lokasi . jika hal ini terjadi, arusnya berbalik untuk merubah kutub – kutub utara dan selatan dinamo.

3. Komutator

Komponen ini terutama ditemukan dalam motor DC. Kegunaanya adalah untuk membalikan arah arus listrik dalam dinamo. Komutator juga membantu dalam transmisi arus antara dinamo dan sumber daya.

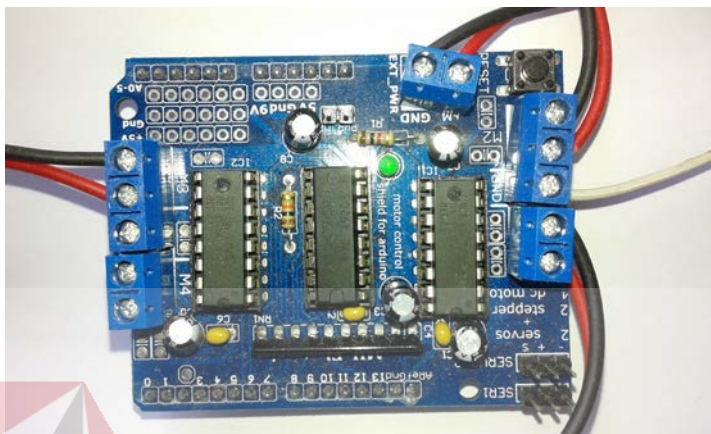
Keuntungan utama motor DC adalah sebagai pengendali kecepatan, yang tidak mempengaruhi kualitas pasokan daya. Motor DC dapat dikendalikan dengan cara mengatur tegangan dinamo, karena meningkatkan tegangan dinamo akan meningkatkan kecepatan. Mengatur arus medan karena menurunkan arus medan akan meningkatkan kecepatan. Motor Dc tersedia dalam banyak ukuran, namun penggunaannya pada umumnya dibatasi untuk beberapa pengguna berkecepatan rendah, penggunaan daya rendah hingga sedang seperti peralatan mesin dan *rolling mills*, sebab sering terjadi masalah dengan perubahan arah arus listrik mekanis pada ukuran yang lebih besar. Motor tersebut dibatasi hanya untuk penggunaan di area yang bersih dan tidak berbahaya sebab resiko percikan api pada sikatnya.

2.6 Shield Motor Driver

Motor Driver Shield untuk Arduino ini dirancang berdasarkan rancangan Adafruit Motor Driver shield yang populer dan sangat berguna untuk mengendalikan berbagai macam motor dari Arduino. Cukup dengan satu *shield* (modul elektronika yang dirancang khusus untuk Arduino dengan bentuk papan dan posisi pin standar), Dengan menggunakan shield module ini, Arduino Anda dapat mengendalikan dua servo motor atau dua stepper motor atau empat DC motor sekaligus.

Arduino Shield ini menggunakan dua sirkuit terpadu L293D yang merupakan Quadruple Half H-Bridge Driver / *Push-Pull Four Channel Driver* yang sudah dilengkapi dengan jepitan dioda (*diode clamps*) sebagai mekanisme proteksi terpadu pada setiap kanalnya. Pengendalian dapat dilakukan cukup

dengan 3-pin dari Arduino karena modul ini dilengkapi dengan IC 74HC595 yang merupakan IC Shift Register yang "mengembangkan" 3-pin serial tersinkronisasi menjadi 8 pin digital masukan pengendali pada kedua IC L293D (4 pin masukan untuk masing-masing IC).



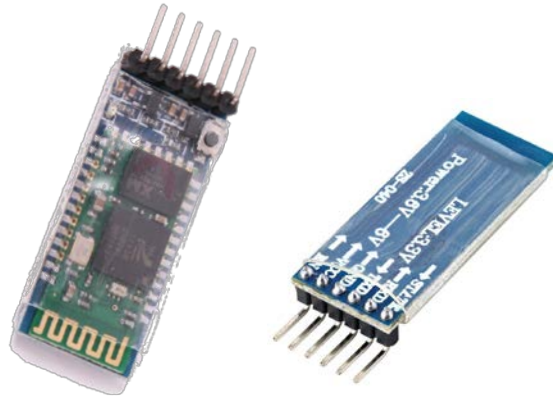
Gambar 2.8 Tampilan Shield Motor Driver

2.7 Modul Bluetooth HC-05

Bluetooth Module HC-05 merupakan module komunikasi nirkabel pada frekuensi 2.4GHz dengan pilihan koneksi bisa sebagai slave, ataupun sebagai master. Sangat mudah digunakan dengan mikrokontroler untuk membuat aplikasi wireless. Interface yang digunakan adalah serial RXD, TXD, VCC dan GND. Built in LED sebagai indikator koneksi bluetooth.

Tegangan input antara 3.6 ~ 6V, jangan menghubungkan dengan sumber daya lebih dari 7V. Arus saat unpaired sekitar 30mA, dan saat paired (terhubung) sebesar 10mA. 4 pin interface 3.3V dapat langsung dihubungkan ke berbagai macam mikrokontroler (khusus Arduino, 8051, 8535, AVR, PIC, ARM, MSP430,

etc.). Jarak efektif jangkauan sebesar 10 meter, meskipun dapat mencapai lebih dari 10 meter, namun kualitas koneksi makin berkurang.



Gambar 2.9 Tampilan Bluetooth HC-05 6 Pins

Spesifikasi

- **Bluetooth protocol:** Bluetooth Specification v2.0+EDR
- **Frequency:** 2.4GHz ISM band
- **Modulation:** GFSK(Gaussian Frequency Shift Keying)
- **Emission power:** 4dBm, Class 2
- **Sensitivity:** -84dBm at 0.1% BER
- **Speed:** Asynchronous: 2.1Mbps(Max) / 160 kbps, Synchronous: 1Mbps/1Mbps
- **Security:** Authentication and encryption
- **Profiles:** Bluetooth serial port
- **Power supply:** +3.3VDC 50mA
- **Working temperature:** -20 ~ +75 Centigrade
- **Dim**
- **ension:** 3.57cm x 1.52cm

Kelebihan menggunakan Bluetooth

- Media komunikasi yang praktis karena tidak membutuhkan kabel
- Dapat menembus beberapa rintangan seperti halnya dinding, kayu dan lain-lain
- Dapat melakukan sinkronisasi data dari perangkat mobile dengan komputer atau laptop

Kelemahan Menggunakan Bluetooth

- Jarak komunikasi terbatas
- Menggunakan frekuensi yang sama dengan sinyal wifi
- Banyak virus yang tersebar melalui media bluetooth

Kecepatan transfer tidak stabil karena tergantung dari perangkat pengirim dan penerima

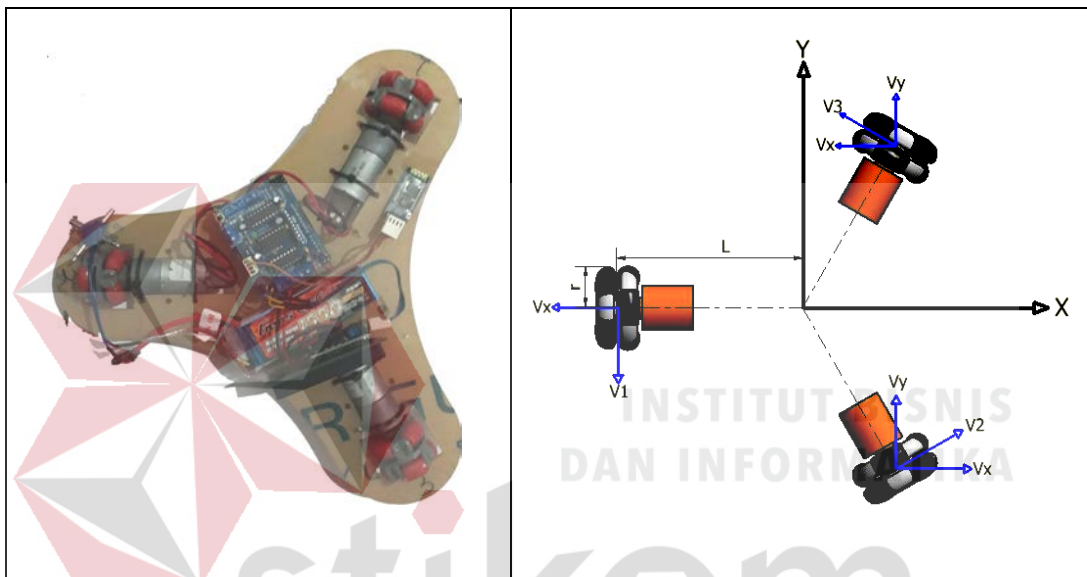
2.8 Roda Omni Directional

Roda omni directional adalah rancangan roda khusus yang hanya memiliki roda tunggal, namun terdiri dari roda inti berukuran besar dan diantaranya terdapat roda – roda kecil dengan poros tegak lurus pada roda inti

Konsep pergerakan roda tersebut digunakan agar dapat bergerak kesegala arah, dengan kemampuan 10 arah gerak berbeda, mulai dari maju, mundur, putar kanan, putar kiri, geser kanan, geser kiri, serong kanan atas, serong kanan bawah, serong kiri atas, dan serong kiri bawah, untuk mencapai pergerakan tersebut maka kita membutuhkan perhitungan kinematika dari serta memiliki persamaan kinematika dalam pergerakannya yang dirumuskan.

2.8.1 Persamaan Kinematika 3 Roda Omni-Directional

Dari perkembangan roda omni directional robot tersebut dapat diteliti dari sistem kinematika untuk pergerakan atau perpindahan terhadap posisi awalnya, dengan mengetahui variabel – variabel yang berkaitan satu dengan lainnya yang dapat dikendalikan pada masing – masing roda, untuk menggerakkan robot pada posisi tertentu.



Gambar 2.10 Kinematika Tiga Roda Omni

Dimana :

- X, Y = Sistem Koordinat Kartesian.
- V_x, V_y = Kecepatan roda ke sumbu X, Y
- $V_i(1,2,3)$ = Vektor Putaran dari masing-masing roda omni.
- L = Jarak Pusat Robot ke titik pusat roda.
- r = Jari-jari dari roda omni

Pergerakan dari roda omni memiliki 2 DOF pergerakan, dimana dapat bergerak rotasi sesuai dengan putaran motor dan bergerak translasi karena pengaruh dari dorongan roda lainnya yang diasumsikan memiliki kecepatan V_x ,

V_y dan V_θ terhadap titik referensi koordinat kartesian. Hal ini dapat digunakan untuk menginvestigasi kecepatan dari ketiga roda robot yang telah di rangkai menjadi satu seperti yang ditunjukkan pada Gambar 3.0 di atas, untuk mendapatkan kecepatan dari roda robot (V_i) dapat kita turunkan untuk mendapatkan kecepatan ketiga roda robot pada bidang koordinat kartesian robot. Untuk mendapatkan kecepatan dari masing-masing dari roda omni sama dengan kecepatan dari motor yang dikalikan dengan jari-jari dari roda omni yang dirangkai secara simetris dengan perbedaan sudut antar roda sebesar 120° . Maka masing-masing roda memiliki sudut θ sebesar 60° , sehingga jika diturunkan ketiga roda omni memiliki rumus sebagai berikut:

$$V_x = (V_1 \cos 30 - V_2 \sin 30 + V_3 L) \quad (1)$$

$$V_y = (V_1 \sin 30 + V_2 \cos 30 + V_3 L) \quad (2)$$

$$V_\theta = (-V_1 \sin 30 + V_2 \cos 30 + V_3 L) \quad (3)$$

Untuk menentukan kecepatan dari robot dan orientasi arah hadap robot dapat menggunakan rumus sebagai berikut:

$$V = \sqrt{V_x^2 + V_y^2} \quad (4)$$

$$\theta = \tan^{-1} \frac{V_y}{V_x} \quad (5)$$