

BAB II

LANDASAN TEORI

2.1 *Traverser*

Traverser adalah sebuah alat bantu penyeberangan material dan badan kereta api untuk dipindahkan dari suatu proses ke proses selanjutnya. *Traverser* bisa dilihat pada gambar.

Traverser pada gambar 1 adalah salah satu *Traverser* dengan ukuran yang besar yang dimiliki PT. Industri Kereta Api (Persero). *Traverser* memiliki dimensi panjang 29236mm dan lebar 6000mm. Pada *Traverser* terdapat ruang operator dan 2 alat penyambung *track* di kanan dan di kiri.



Gambar 2.1 *Traverser*

Mempunyai 7 buah motor DC sebagai penggerak dengan diberi tenaga oleh generator diesel bertegangan 220v AC dan beban terberat yang pernah diangkut kurang lebih 100 ton. *Traverser* ini secara umum mempunyai 3 operator,

2 operator bertugas mengoperasikan alat penyambungan *track* dan 1 operator sebagai penggerak *Traverser*.

2.2 Bogie

Bogie merupakan sistem kesatuan roda pada kereta api, baik di kereta berpenggerak maupun kereta non penggerak. *Bogie* pada umumnya dipakai untuk roda yang jumlahnya lebih dari 2 gandar (As) dalam satu kereta.

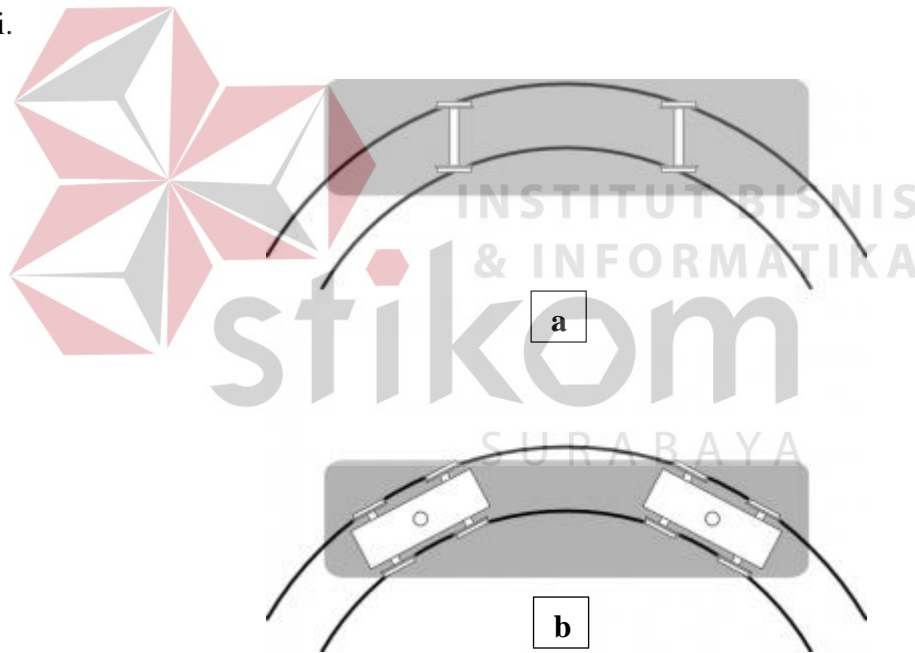


Gambar 2.2 Kereta Gandar 2 dan Kereta ber-*Bogie*

Bogie adalah suatu konstruksi yang terdiri dari dua perangkat roda atau lebih yang digabungkan oleh rangka yang dilengkapi dengan sistem pemegasan, pengereman, dengan atau tanpa peralatan penggerak dan anti selip, serta keseluruhan berfungsi sebagai pendukung rangka dasar dari badan kereta. *Bogie* dapat dilepas dan dipasang kembali jika sedang dilakukan perawatan. Dapat dilihat pada gambar 2.2 diatas untuk membedakan kereta bergandar dan ber-*Bogie*.

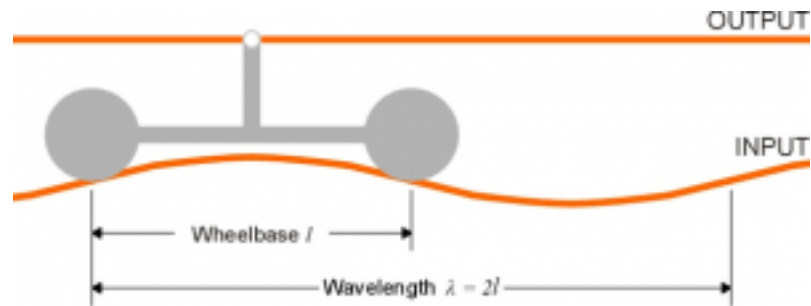
Fungsi utama *Bogie* adalah menghasilkan fleksibilitas kereta terhadap rel sehingga roda dapat tetap mengikuti arah rel saat melewati tikungan. Saat kereta melewati rel yang membelok atau menikung, maka akan terjadi sudut antara garis

lurus badan kereta dengan rel. Pada keadaan ini, akan terjadi kontak antara *flens* dengan rel pada salah satu sisi rodanya. Pada kereta tanpa *Bogie* maka sudut ini terbatas karena roda akan selalu segaris dengan badan kereta sehingga saat *flens* sudah tidak bisa menahan rel, maka roda akan naik ke atas rel dan akhirnya terjadi *derailment* atau anjlok. Dengan adanya *Bogie*, maka roda tidak segaris dengan badan kereta melainkan mempunyai sudut tertentu yang memungkinkan roda bisa membelok mengikuti rel tanpa terjadi anjlok atau roda yang naik ke atas rel. Untuk lebih jelasnya dapat dilihat pada gambar 2.3 dibawah ini.



Gambar 2.3 *Bogie Steering Tanpa Bogie (a), Bogie Steering Dengan Bogie (b)*

Selain fleksibilitas, *Bogie* juga dapat meredam efek yang diakibatkan oleh rel yang bergelombang naik turun. Titik tengah *Bogie* yang disebut *center pivot* akan membagi defleksi yang terjadi diantara 2 rodanya.



Gambar 2.4 Peredaman Ketidakrataan Rel Oleh *Bogie*

Hal ini akan menyebabkan kereta lebih stabil walau rel tidak rata/bergelombang naik turun. Peredaman *Bogie* melintasi ketidakrataan rel dapat dilihat pada gambar 2.4.



Gambar 2.5 *Temporary Bogie* (<http://www.inka.co.id/?p=2178>)

Temporary Bogie merupakan bentuk dari sebuah *Bogie* biasa, hanya saja *Temporary Bogie* dibuat untuk sementara saja dalam artian berguna untuk

menopang atau membawa bahan material seperti plat besi, roda kereta, *frame* kereta dan lain-lain. Pada dasarnya *Temporary Bogie* hanya berbentuk *frame body* sederhana dengan penopang, *axle box* dan 2 pasang roda. Di PT. Industri Kereta Api (Persero) sendiri *Temporary Bogie* digunakan untuk memindahkan material barang maupun badan gerbong antar *workshop*. (Gambar 2.5) diatas merupakan contoh *Temporary Bogie* yang dipakai untuk mengangkat bahan atau material yang berat.

2.3 Arduino UNO

Arduino UNO adalah sebuah board perangkat *prototype* elektronik berbasis mikrokontroler yang fleksibel dan *open-source*. Perangkat keras dan perangkat lunaknya mudah dipelajari dan digunakan, Arduino UNO bisa diimplementasikan pada pendeteksi lingkungan dengan *input* beberapa sensor (cahaya, suhu, *infrared*, ultrasonik, jarak, tekanan, kelembaban), dan dapat mengendalikan peralatan sekitarnya (seperti: lampu, motor, aktuator, dll.). Board berbasis ATmega 328P yang memiliki 14 pin *digital input/output* pin (dimana 6 pin sebagai PWM) 6 input, Analog, 16 MHz osilator kristal, jack listrik, koneksi USB, tombol reset, ICSP *header*. Hampir semua yang diperlukan untuk mendukung mikrokontroler sudah ada. Penggunaan hanya menghubungkan kabel USB ke komputer atau kekuasaan itu dengan adaptor AC-DC atau baterai untuk powernya. Kebutuhan mikrokontroler semua didukung pada perangkat ini. Arduino UNO ini berbeda dengan papan-papan Arduino yang lain, di mana pada versi-versi Arduino sebelumnya digunakan *chip* FTDI *USB-to-serial*, namun pada

Arduino UNO digunakan ATmega8U2 yang diprogram sebagai *converter USB-to-serial*. Kata “UNO” merupakan bahasa Italia yang artinya adalah satu, dan diberi nama demikian sebagai penanda peluncuran Arduino 1.0. Arduino UNO merupakan versi yang paling baru hingga saat ini dari kelompok papan 12. Arduino USB.

Arduino UNO bersama dengan Arduino 1.0 selanjutnya menjadi acuan untuk pengembangan Arduino versi selanjutnya. Arduino UNO mempunyai beberapa fasilitas untuk dapat berkomunikasi dengan komputer, Arduino lain, atau dengan mikrokontroler lain. Mikrokontroler ATmega328P pada Arduino UNO menyediakan komunikasi serial UART TTL (5 V), yang tersedia pada pin 0 (RX) dan 1 (TX). ATmega8U2 pada papan Arduino UNO menyalurkan komunikasi serial ini melalui USB dan hadir sebagai *com port virtual* pada *software* di komputer. *Firmware* dari ATmega8U2 menggunakan *driver* USB COM standar, dan tidak dibutuhkan *driver* eksternal. *Software* Arduino memiliki serial monitor yang memungkinkan data teks sederhana dikirim ke dan dari Arduino. LED RX dan TX akan berkedip ketika data sedang ditransmisikan melalui *chip USB-to-serial*. ATmega328P juga mendukung komunikasi I2C (TWI) dan SPI. *Software* Arduino mempunyai *library* *Wire* dan SPI untuk menyederhanakan penggunaan bus I2C dan komunikasi SPI.

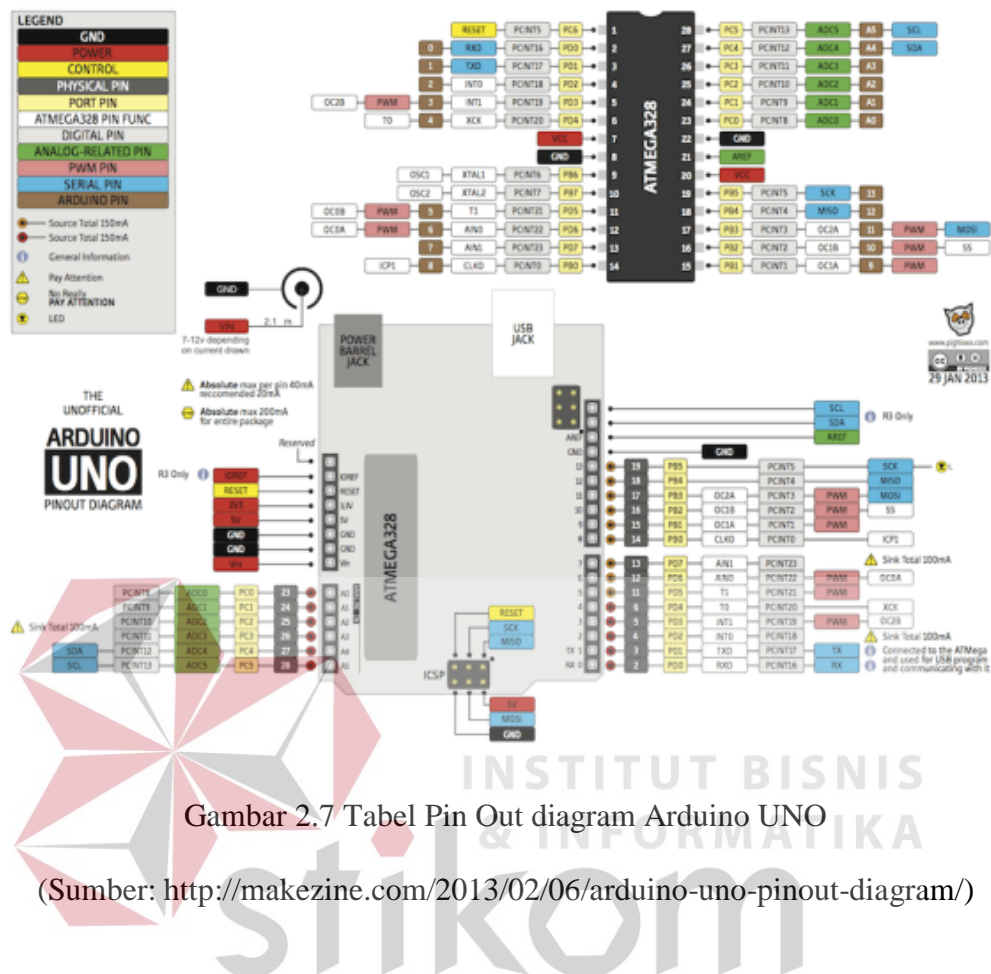
2.3.1 Hardware

Dibawah ini merupakan bentuk fisik *board* Arduino UNO yang digunakan untuk bahan Tugas Akhir ini lengkap dengan tabel *port* I/O beserta

dengan keterangan yang dilengkapi dengan warna pembeda di setiap *port* I/O yang ada pada Arduino UNO dapat dilihat pada (Gambar 2.6) dan (Gambar 2.7) berikut ini :



Gambar 2.6 Board Arduino UNO



Gambar 2.7 Tabel Pin Out diagram Arduino UNO

(Sumber: <http://makezine.com/2013/02/06/arduino-uno-pinout-diagram/>)

Berikut Spesifikasi *board* Arduino Uno dapat dilihat pada Tabel 2.1

berikut ini:

Tabel 2.1 Spesifikasi Arduino Uno

| | |
|-----------------------------|------------------------------------|
| Microcontroller | Atmega328P |
| Operating Voltage | 5 V |
| Input Voltage (recommended) | 7 - 12 V |
| Input Voltage (limit) | 6 - 20 V |
| Digital I/O Pins | 14 (of which 6 provide PWM Output) |

| | |
|--------------------------|---|
| PWM Digital I/O Pins | 6 |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 20 mA |
| DC Current for 3.3 V Pin | 50 mA |
| Flash Memory | 32 KB (Atmega328P) of which 0.5 KB used by bootloader |
| SRAM | 2 KB (Atmega328P) |
| EEPROM | 1 KB (Atmega328P) |
| Clock Speed | 16 MHz |
| Length | 68.6 mm |
| Width | 53.4 mm |
| Weight | 25 g |

1. Memori

ATmega328P memiliki *Flash Memory* 32 KB (0,5 KB digunakan untuk *bootloader*), 2 KB SRAM dan 1 KB EEPROM (yang dapat dibaca dan ditulis dengan *library* EEPROM). (Arduino.cc)

2. Input – Output

Masing-masing dari 14 digital pin (pin *header*) pada Arduino Uno dapat digunakan sebagai *input* atau *output*, dengan menggunakan fungsi dari `pinMode()`, `digitalWrite()`, dan `digitalRead()`, serta beroperasi pada tegangan 5 V. Setiap pin dapat memberikan atau menerima arus maksimum 20 mA dan memiliki resistor

pull-up internal yang (terputus secara *default*) dari 20-50k Ohm. Selain itu, beberapa pin memiliki fungsi khusus:

- **Serial:** 0(RX) dan 1(TX) yang digunakan untuk menerima (RX) dan mengirimkan (TX) TTL data serial. Pin juga terhubung ke pin yang sesuai dari ATmega16U2 *USB-to-TTL* chip Serial.
- **Eksternal Interupsi:** Pin 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu *interrupt* pada nilai yang rendah, naik atau jatuh tepi, atau perubahan nilai.
- **PWM:** Pin 3, 5, 6, 9, 10, 11. Menyediakan 8-bit PWM *output* dengan `analogWrite ()` function.
- **SPI (Serial Peripheral Intervace):** 10(SS), 11(MOSI), 12(MISO), 13(SCK). Pin ini mendukung komunikasi SPI menggunakan *library* SPI.
- **LED:** 13. Ada *built-in* LED terhubung ke pin digital 13. Ketika pin dengan nilai TINGGI, LED menyala, ketika pin yang dipakai RENDAH, lampu akan mati.
- **TWI (Two-Ware Inteerface):** Pin A4(SDA) dan pin A5(SCL). Dukungan komunikasi TWI menggunakan *library* Wire.
- Arduino Uno memiliki 6 *input* analog, dengan label A0 hingga A5, yang masing-masing menyediakan 10 bit resolusi (yaitu 1024 nilai yang berbeda). Secara *default* mereka mengukur dari *ground* sampai 5 V, meskipun mungkin untuk mengubah jangkauan menggunakan pin AREF dan fungsi `analogReference ()`. (Arduino.cc)

3. Daya (Power)

Arduino Uno dapat diaktifkan melalui koneksi USB atau dengan catu daya *eksternal*. Sumber daya dipilih secara otomatis. Eksternal (*non-USB*) daya dapat berasal baik dari adaptor AC-DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan *plug* ukuran 2.1 mm pusat-positif ke colokan listrik *board*. Baterai dapat dimasukkan dalam Gnd dan Vin pin *header* dari konektor daya.

Board dapat beroperasi pada pasokan eksternal 6 sampai 20 V. Jika tegangan kurang dari 7 V, tegangan pada *board* kemungkinan akan tidak stabil. Jika menggunakan lebih dari 12 V, regulator tegangan bisa panas dan merusak *board*. Kisaran yang disarankan adalah 7 sampai 12 V.

Pin daya adalah sebagai berikut:

- **VIN:** Tegangan *input* ke *board* Arduino ketika menggunakan sumber daya eksternal (sebagai lawan 5 V dari koneksi USB atau sumber daya diatur lain). Tegangan dapat dipasok melalui pin ini.
- **5 V:** Pin *output* 5 V diatur dari regulator di *board*. *Board* dapat diaktifkan dengan daya baik dari colokan listrik DC (7-12 V) , konektor USB (5 V), atau pin VIN dari *board* (7-12 V). Jika tegangan diberikan melalui 5 V atau 3.3 V melewati regulator, dan dapat merusak *board*, maka tidak disarankan.
- **3.3 V:** Sebuah pasokan 3.3 V dihasilkan oleh regulator *on-board* yang dapat menarik arus maksimum 50 mA.
- **GND:** Pin tanah.

- **IOREF:** Pin pada board Arduino memberikan tegangan referensi saat mikrokontroler sedang beroperasi. Sebuah shield dikonfigurasi dengan benar agar dapat membaca pin tegangan IOREF dan memilih sumber daya yang tepat atau mengaktifkan penerjemah tegangan pada output untuk bekerja dengan tegangan 5 V atau 3.3 V. (arduino.cc)

2.3.2 Software

Arduino IDE adalah sebuah *software* pengendali yang digunakan untuk menulis program, mengcompile, dan mengunggah ke papan Arduino. *Arduino Development Environment* terdiri dari editor teks untuk menulis kode, area pesan, konsole teks, toolbar dengan tombol-tombol untuk fungsi umum, dan sederet menu. *Software* ini bersifat *open-sources*, diturunkan dari *platform wiring*, dirancang untuk memudahkan *user*.



Gambar 2.8 Tampilan IDE Pada Arduino

Bahasa pemrograman ini adalah C++ yang sederhana dan fungsinya yang lengkap sehingga Arduino mudah dipelajari. Ada beberapa menu bar seperti pada umumnya meliputi *file*, *edit*, *sketch*, *tools*, *help*. Instalasi *software* ini bisa secara manual dan online di akses pada website:// www.arduino.cc. *Software* yang ditulis menggunakan Arduino dinamakan *sketches*. *Sketches* ini ditulis di editor teks dan disimpan dengan *file* yang berekstensi *.ino*. Editor teks ini mempunyai fasilitas untuk *cut/paste* dan

search/replace. Area pesan berisi umpan balik ketika menyimpan dan mengunggah *file*, dan juga menunjukkan jika terjadi *error*. Tampilan IDE Arduino dapat dilihat pada gambar 2.8.

A. Beberapa pilihan menu file:

- *New* : Membuat *Sketch* baru.
- *Open* : membuka file sketch yang sudah tersimpan.
- *Sketchbook* : membuat file sketch yang pernah dibuat.
- *Examples* : membuka contoh-contoh *file Sketch* yang berisi berbagai macam aplikasi yang disediakan *platform* Arduino.
- *Close* : menutup *Sketch*.
- *Save* : menyimpan *sketch*.
- *Save As* : Menyimpan *Sketch* dengan nama lain.
- *Upload to I/O Board* : mengunggah program ke ke *board*.
- *Page setup* : mengatur halaman pada pencetak.
- *Print* : mencetak *sketch*.
- *Preference* : Mengatur setting IDE Arduino
- *Quit* : keluar dari IDE Arduino.

B. Beberapa Menu Sketch:

- *Verify / compile* : Mengompilasi program.
- *Stop* : menghentikan kompilasi (apabila *Hang*).
- *Show sketch folder* : menampilkan *folder I* dari *sketch* yang sedang dibuka.
- *Import Library* : mengambil *header* library dari fungsi- fungsi tambahan.
- *Add File* : menambah buka *file sketch* pada jendela yang sama.

C. Bahasa Pemrograman Arduino

Perangkat keras Arduino dapat bekerja secara maksimal dengan bantuan program. Program untuk Arduino berbasis C/C+. Agar program dapat berjalan dengan baik maka perlu setidaknya dua bagian atau fungsi, yaitu:

- a. **Void setup() { ... }** , semua kode di dalam kurung kurawal akan dijalankan hanya satu kali ketika program arduino dijalankan untuk pertama kalinya.
- b. **Void loop() { ... }** , fungsi ini dijalankan setelah *setup* (fungsi *void setup*) selesai. Setelah dijalankan satu kali fungsi ini akan dijalankan kembali, dan lagi secara terus menerus sampai catu daya (*power*) dilepaskan.

Syntax, berikut ini adalah elemen bahasa c yang dibutuhkan untuk format penulisan:

- a. **// (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 di belakangnya akan diabaikan oleh program.
- b. **/* (komentar banyak baris)**, jika Anda mempunyai banyak catatan, maka hal tersebut dapat dituliskan pada beberapa baris sebagai komentar. Semua hal yang terletak di antara dua simbol tersebut akan diabaikan oleh program.
- c. **{ ... } atau kurung kurawal**, digunakan untuk mendefinisikan kapan blok program mulai dan berakhir (digunakan juga pada fungsi dan pengulangan).
- d. **;** (**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 didefinisikan sebagai instruksi untuk memindahkan angka dengan cara yang cerdas. Variabel inilah yang digunakan untuk memudahkannya.

- a. **Int (integer)**, digunakan untuk menyimpan angka dalam 2 byte (16 bit). Tidak mempunyai angka desimal dan menyimpan nilai dari -23.768 s/d 32.767.
- b. **Long**, digunakan ketika integer tidak mencukupi lagi. Memakai 4 byte (32 bit) dari memori RAM dan mempunyai rentang nilai dari -2.147.648 s/d 2.147.483.647.
- c. **Boolean**, variabel sederhana yang digunakan untuk menyimpan nilai *TRUE* (benar) atau *FALSE* (salah). Sangat berguna karena hanya menggunakan 1 bit dari RAM.
- d. **Float**, digunakan untuk angka desimal (*floating point*). Memakai 4 byte (32 bit) dari RAM dan mempunyai rentang nilai dari -3,4028235E+38 s/d 3,4028235E+38.
- e. **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).

- a. **= (sama dengan)**, membuat sesuatu menjadi sama dengan nilai yang lain (misalnya: $x = 10 * 2$, $x = 20$).
- b. **% (persen)**, menghasilkan sisa dari hasil pembagian suatu angka yang lain (misalnya : $12 \% 10$, ini akan menghasilkan angka 2).
- c. **+ (plus)**, operasi penjumlahan.
- d. **- (minus)**, operasi pengurangan.
- e. *** (asteris)**, operasi perkalian.
- f. **/ (garis miring)**, operasi pembagian.

Operator Pembandingan, digunakan untuk membandingkan nilai logika.

- a. **== (sama dengan)**, misalnya: $12 == 12$ adalah *TRUE* (benar).
- b. **!= (tidak sama dengan)**, misalnya: $12 != 10$ adalah *TRUE* (benar) atau $12 != 12$ adalah *FALSE* (salah).
- c. **< (lebih kecil dari)**, misalnya: $12 < 10$ adalah *FALSE* (salah) atau $12 < 12$ adalah *FALSE* (salah) atau $12 < 14$ adalah *TRUE* (benar).
- d. **> (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.

- a. **If ... else**, dengan format seperti berikut ini :

```
If (kondisi) { ... }
Else if (kondisi) { ... } Else
{ ... }
```

Dengan struktur seperti di atas 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.

- b. **For**, dengan format penulisan sebagai berikut :

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

Digunakan bila Anda ingin melakukan pengulangan kode program di dalam kurung kurawal beberapa kali, ganti #pengulangan dengan jumlah

pengulangan yang diinginkan. Melakukan perhitungan ke atas (++) atau ke bawah (--).

Digital

a. **pinMode(pin, mode)**, digunakan untuk menetapkan mode dari suatu pin, pin adalah nomor pin yang akan digunakan sebagai *port* dari 0 s/d 19 (pin analog 0 s/d 5 adalah 14 s/d 19). Mode yang bisa digunakan adalah *INPUT* atau *OUTPUT*.

b. **digitalWrite(pin, value)**, ketika sebuah pin ditetapkan sebagai *OUTPUT*, pin tersebut dapat dijadikan *HIGH* (+5 V) atau *LOW* (*ground*).

c. **digitalRead(pin)**, ketika sebuah pin ditetapkan sebagai *INPUT*, maka Anda dapat menggunakan kode ini untuk mendapatkan nilai pin tersebut apakah *HIGH* (+5 V) atau *LOW* (*ground*).

Analog, arduino adalah mesin digital tetapi mempunyai kemampuan untuk beroperasi di dalam analog.

a. **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* ~ 0 V) dan 255 (100% *duty cycle* ~ 5 V).

b. **analogRead(pin)**, ketika pin analog ditetapkan sebagai *INPUT* Anda dapat membaca keluaran voltase-nya. Keluarannya berupa angka antara 0 (untuk 0 V) dan 1024 (untuk 5 V).

2.4 LCD

LCD (*Liquid Crystal Display*) adalah suatu jenis media tampil yang menggunakan kristal cair sebagai penampil utama. LCD digunakan diberbagai bidang seperti alat elektronik, kalkulator, atau pun layar komputer. LCD menggunakan dot matrik dengan jumlah karakter 2 x 16 sebagai media penampil dan dapat dilihat pada (Gambar 2.9). LCD berfungsi sebagai penampil yang akan digunakan untuk menampilkan status kerja alat. LCD ini memerlukan tiga jalur kontrol dan delapan jalur data (untuk mode 8 bit) serta empat jalur data (untuk mode 4 bit). Ketiga jalur kontrol yang dimaksud adalah pin EN, RS dan RW.

EN adalah pin *Enable* dimana jalur ini digunakan untuk memberitahu LCD kalau kita akan berkomunikasi dengannya. Sebelum mengirim data ke LCD jalur ini di buat berlogika tinggi dahulu. Kemudian jalur kontrol yang lain di *setting*, pada saat bersamaan data yang akan dikirim ditempatkan pada jalur data. Setelah semua siap, jalur EN dibuat berlogika rendah. Transisi dari logika tinggi ke logika rendah ini akan memberitahu LCD untuk mengambil data pada jalur kontrol dan jalur data.

RS adalah pin *Register select*. Pada saat pin RS berlogika rendah, data yang dikirim adalah perintah-perintah seperti membersihkan layar, posisi kursor, dan lain-lain. Sedangkan jika berlogika tinggi data yang dikirim adalah teks data di mana teks ini yang harus ditampilkan pada layar.



Gambar 2.9 LDC 2x16

RW adalah pin *Read/Write*. Pada saat pin RW berlogika rendah, informasi pada jalur data berupa pengiriman data ke LCD (*write*). Sedangkan ketika pin RW berlogika tinggi, berarti sedang dilaksanakan pengambilan data dari LCD (*read*). Sedangkan untuk jalur data terdiri dari delapan bit, data ini disebut D0, D1, D2, D3, D4, D5, D6 dan D7 seperti yang ditunjukkan pada Tabel 2.2 dibawah ini :

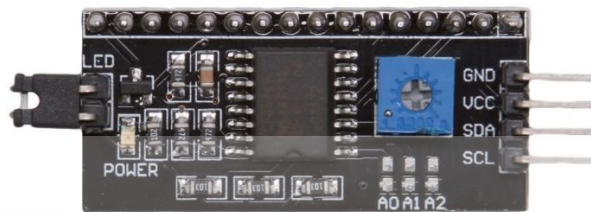
Tabel 2.2 Menunjukkan Fungsi dan Konfigurasi Pin LCD 16X2 (Syahrul, 2014)

| Pin | Nama | Fungsi |
|-----|------|---|
| 1 | VSS | Ground |
| 2 | VCC | +5V |
| 3 | VEE | Tegangan kontras |
| 4 | RS | Register Select (0=Register instruksi, 1=Register data) |
| 5 | R/W | Untuk memilih mode tulis atau baca (0=tulis, 1=baca) |
| 6 | E | Enable (0=enable/menahan data ke LCD, 1=disable) |
| 7 | DB0 | Data Bit 0, LSB |
| 8 | DB1 | Data Bit 1 |
| 9 | DB2 | Data Bit 2 |
| 10 | DB3 | Data Bit 3 |
| 11 | DB4 | Data Bit 4 |
| 12 | DB5 | Data Bit 5 |
| 13 | DB6 | Data Bit 6 |
| 14 | DB7 | Data Bit 7 |
| 15 | BPL | Back Plane Light |
| 16 | GND | Ground |

I2C (*Inter Integrated Circuit*)

I2C (Inter Integrate Circuit) adalah standar komunikasi serial dua arah menggunakan dua saluran yang didesain khusus untuk pengontrolan IC (Lihat

Gambar 2.10). Sistem I2C terdiri dari saluran SCL (*Serial Clock*) dan SDA (*Serial Data*) yang membawa informasi data antara I2C dengan pengontrolnya. Piranti yang dihubungkan dengan sistem I2C Bus dapat dioperasikan sebagai Master dan Slave. Master adalah piranti yang memulai transfer data pada I2C Bus dengan membentuk sinyal Start, mengakhiri transfer data dengan membentuk sinyal Stop, dan membangkitkan sinyal clock. Slave adalah piranti yang dialamati master.



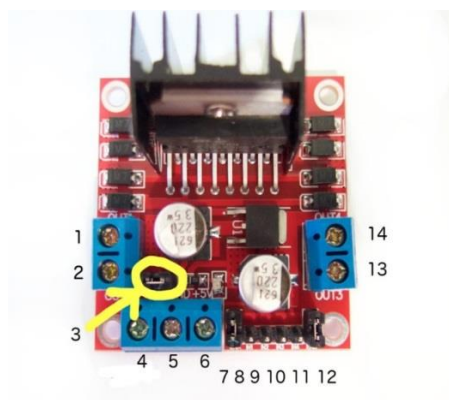
Gambar 2.10 I2C/SPI LCD Backpack

Sinyal Start merupakan sinyal untuk memulai semua perintah, didefinisikan sebagai perubahan tegangan SDA dari “1” menjadi “0” pada saat SCL “1”. Sinyal Stop merupakan sinyal untuk mengakhiri semua perintah, didefinisikan sebagai perubahan tegangan SDA dari “0” menjadi “1” pada saat SCL “1”. Kondisi sinyal Start dan sinyal Stop. Sinyal dasar yang lain dalam I2C Bus adalah sinyal acknowledge yang disimbolkan dengan ACK. Setelah transfer data oleh master berhasil diterima slave, slave akan menjawabnya dengan mengirim sinyal acknowledge, yaitu dengan membuat SDA menjadi “0” selama siklus clock ke-9. Ini menunjukkan bahwa Slave telah menerima 8 bit data dari Master.

1. Hanya melibatkan 2 kabel yaitu serial data line.
2. Setiap IC yang terhubung dalam I2C memiliki alamat masing-masing yang dapat diatur secara Software dengan master/slave protocol yang sederhana, dan mampu mengakomodasikan multi master.

2.5 Modul Motor Driver L298 Dual H-Bridge 2A

Modul *driver* L298N, menggunakan chip ST L298N yang dapat secara langsung mengontrol dua motor DC yang bekerja pada tegangan 3-30V, dan menyediakan antarmuka keluaran 5v DC, juga dapat mengontrol dengan mudah kecepatan motor dan arah pergerakan motor DC yaitu berlawanan arah jarum jam maupun berlawanan arah jarum jam. Modul ini dilengkapi dengan rangkaian sensor arus beban yang dapat digunakan sebagai umpan balik ke pengendali. Modul ini dilengkapi dengan rangkaian sensor arus beban yang dapat digunakan sebagai umpan balik ke pengendali. Modul ini mampu men-*drive* beban-beban induktif seperti misalnya motor DC dan motor *stepper*. Lihat (Gambar 2.11) modul motor driver L298 dual H-Bridge 2A.



Gambar 2.11 Modul Motor Driver L298N Dual H-Bridge 2A

2.5.1 Keterangan gambar L298N dual H-bridge 2A

Tabel 2.3 Keterangan Pin Modul Motor Driver L298N Dual H-Bridge 2A

| No. Pin | I/O | Keterangan |
|---------|-----|--|
| 1 | O | DC motor 1 “+” atau stepper motor A+ |
| 2 | O | DC motor 1 “-” atau stepper motor A- |
| 3 | – | 12V jumper – lepaskan jumper ini jika menggunakan sumber lebih dari 12 V DC. Ini memungkinkan sumber dari regulator pada Arduino 5V. |
| 4 | – | Hubungkan sumber tegangan motor disini, maksimum 35 V DC. Lepaskan 12V jumper V jika >12V. |
| 5 | – | GND |
| 6 | O | 5V output jika 12V jumper digunakan, ideal untuk mensuplai Arduino Anda dll. |
| 7 | I | DC motor 1 enable jumper. Lepaskan ini ketika menggunakan stepper motor. Hubungkan ke keluaran PWM untuk menatur kecepatan motor. |
| 8 | I | IN1 |
| 9 | I | IN2 |
| 10 | I | IN3 |
| 11 | I | IN4 |
| 12 | I | DC motor 2 enable jumper. Lepaskan ini ketika menggunakan stepper motor. Hubungkan ke keluaran PWM untuk menatur kecepatan motor. |
| 13 | O | DC motor 2 “+” or stepper motor B+ |
| 14 | O | DC motor 2 “-” or stepper motor B- |

2.5.2 Spesifikasi Modul Motor Driver L298N Dual H-Bridge 2A

1. Drive Chip: L298N dual H-bridge driver chip.
2. Terminal bagian pengontrol dengan sumber pasokan VMS: +5V~+35V.
3. Bagian Driver dengan Arus maksimal I/O: 2A/Bridge.
4. Bagian Logika dengan terminal *Power Supply* berkisar Vss: 4,5-5,5 V.
5. Bagian Logika dengan arus boperasional berkisar: 0~26mA.
6. Kontrol Tegangan sinyal input berkisar: 4,5-5,5 V low 0V high.
7. Konsumsi power Maksimum: 20W.

8. Temperatur penyimpanan: -25 ~ +130.
9. Ukuran Papan Drive: 55mm*60mm*30mm.
10. Berat papan drive: 33g.
11. Fitur lain: kontrol arah indikator, indikator daya, deteksi arus.

2.6 Infrared Proximity Sensor E18-D80NK

Modul *infrared proximity* E18-D80NK merupakan salah satu jenis dari kategori modul *infrared* dimana modul ini memiliki bentuk yang ringkas dan rapi dari segi pemasangan secara tepat. Untuk mendeteksi ada atau tidaknya benda dibawah robot *Traverser* dan *Bogie*, digunakan sensor jarak yang berbasis pada *infrared* . Modul yang digunakan adalah modul sensor E18-D80NK (Gambar 2.12).



Gambar 2.12 Bentuk Fisik *Infrared Proximity* E18-D80NK

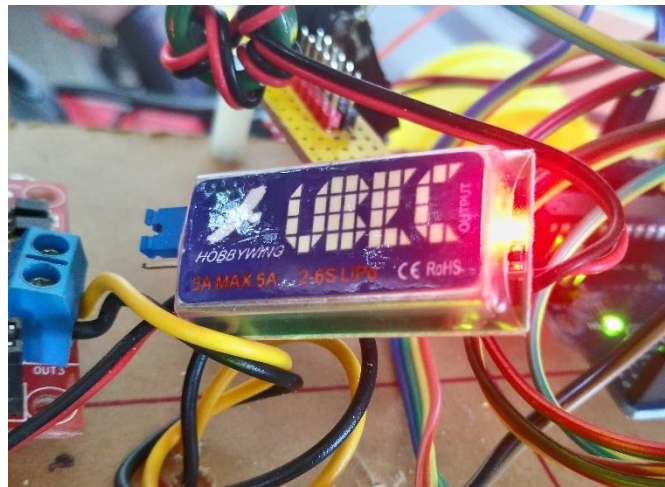
Modul ini mempunyai lensa untuk memfokuskan pengiriman dan penerimaan sinyal inframerah, sehingga modul ini dapat digunakan untuk mendeteksi inframerah sampai pada 80cm. Jangkauan sensor ini dapat diubah

sesuai kebutuhan dengan cara memutar *trimmer* yang ada pada belakang sensor ini. Sensor barang digunakan untuk mendeteksi ada tidaknya barang dihadapan robot. Ketika sensor ini mendeteksi adanya barang di depannya, sensor akan mengirimkan logika 0 pada mikrokontroler. Sensor *Infrared Proximity* E18-D80NK ini memiliki spesifikasi sebagai berikut yaitu:

1. Bertipe reflektif difusi.
2. Sumber cahaya: Infrared.
3. Kisaran Sensing: 3cm ke 80cm (tergantung pada permukaan hambatan) .
4. Tegangan input: 5VDC.
5. Konsumsi arus: 100mA Output.
6. NPN Dimensi: 1.7cm (D) x 4.3cm (L).
7. Panjang kabel: 30 cm.
8. Aplikasi: inframerah sensor menghindari rintangan, sensor jarak, mobil pintar, dll.

2.7 UBEC 3A-5A

UBEC (*Universal Battery Elemination Circuit*) adalah rangkaian elektronik eksternal yang berfungsi memberikan daya dari baterai dan berfungsi sebagai regulasi tegangan hingga 5/6 Volt. UBEC (Gambar 2.13) digunakan pada *Traverser* dan *Bogie* karena spesifikasi UBEC yang menghasilkan tegangan *output* 5/6 VDC yang sesuai kebutuhan pada inputan motor DC, sensor, LCD I2C maupun Instrumentasi lainnya input dan memiliki arus keluaran yang stabil sehingga tidak mudah merusak komponen.



Gambar 2.13 Bentuk fisik UBEC

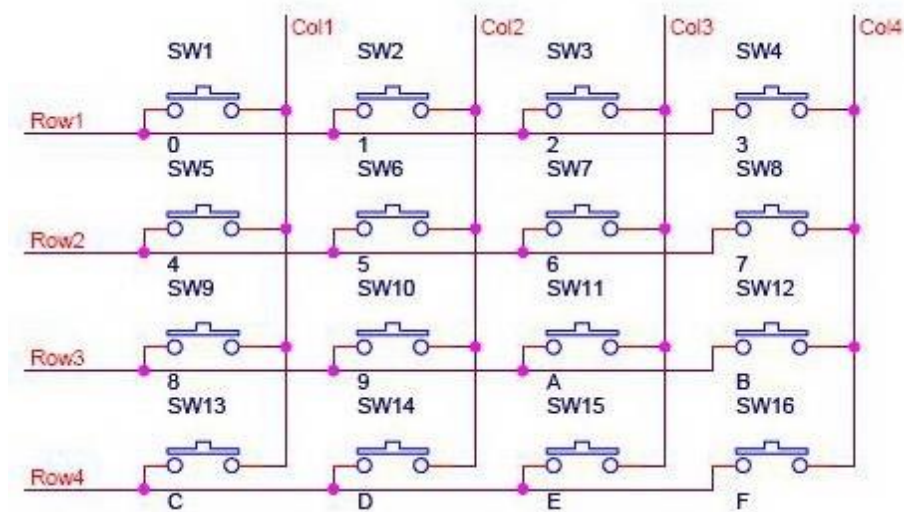
Spesifikasi UBEC sebagai berikut:

1. Input Voltage : 5,5 – 28 V.
2. Output Voltage : 5/6 V.
3. Arus : 25 Ampere.
4. Tipe Regulator : Switching.

INSTITUT BISNIS
& INFORMATIKA
stikom
SURABAYA

2.8 Keypad Membran 4x4

Keypad adalah bagian penting dari suatu perangkat elektronika yang membutuhkan interaksi manusia. Keypad berfungsi sebagai *interface* antara perangkat (mesin) elektronik dengan manusia atau dikenal dengan istilah HMI (*Human Machine Interface*). Matrix keypad 4×4 merupakan salah satu contoh keypad yang dapat digunakan untuk berkomunikasi antara manusia dengan mikrokontroler.



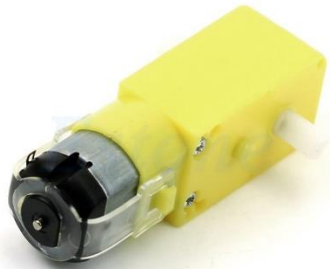
Gambar 2.14 Konstruksi Keypad Membran 4x4

Matrix keypad 4×4 memiliki konstruksi atau susunan yang sederhana dan hemat dalam penggunaan *port* mikrokontroler. Konfigurasi keypad dengan susunan bentuk matrix ini bertujuan untuk penghematan port mikrokontroler karena jumlah *key* (tombol) yang dibutuhkan banyak pada suatu sistem dengan mikrokontroler. Konstruksi matrix keypad 4×4 untuk mikrokontroler dapat dibuat seperti pada (Gambar 2.14). Konstruksi matrix keypad 4×4 diatas cukup sederhana, yaitu terdiri dari 4 baris dan 4 kolom dengan keypad berupa saklar push buton yang diletakan disetiap persilangan kolom dan barisnya. Rangkaian matrix keypad diatas terdiri dari 16 saklar push buton dengan konfigurasi 4 baris dan 4 kolom. Mempunyai 8 line yang terdiri dari 4 baris dan 4 kolom tersebut dihubungkan dengan port mikrokontroler 8 bit. Sisi baris dari matrix keypad ditandai dengan nama *Row1*, *Row2*, *Row3* dan *Row4* kemudian sisi kolom ditandai dengan nama *Col1*, *Col2*, *Col3* dan *Col4*. Sisi input atau output dari

matrix keypad 4×4 ini tidak mengikat, dapat dikonfigurasi kolom sebagai input dan baris sebagai output atau sebaliknya yang bergantung pada *user*-nya.

2.9 Motor DC

Motor DC adalah motor yang bergerak dengan putaran 360 derajat, biasa disebut dinamo, biasa digunakan sebagai penggerak roda, apa bila sumber *positif* dan *negative* dipaang terbalik maka motor DC akan berlawanan arah dari arah putaran sebelumnya. Pada motor DC, daerah kumparan medan yang dialiri arus listrik akan menghasilkan medan magnet yang melingkupi kumparan jangkar dengan arah tertentu. Konversi dari energi listrik menjadi energi mekanik (motor) maupun sebaliknya berlangsung melalui medan magnet, dengan demikian medan magnet di sini selain berfungsi sebagai tempat untuk menyimpan energi, sekaligus sebagai tempat berlangsungnya proses perubahan energi daerah tersebut (Zuhal, 1988). Motor DC yang digunakan adalah Motor DC Reguler yang dilengkapi dengan *gearbox* dengan catu daya maksimal 12 VDC/0,3 A. Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik yang dikombinasikan dengan *gearbox* untuk menaikkan torsi Motor DC (Gambar 2.15). Kumparan medan pada motor dc disebut *stator* (bagian yang tidak berputar) dan kumparan jangkar disebut *rotor* (bagian yang berputar).



Gambar 2.15 Motor DC dengan *Gearbox*

2.10 HMI (*Human Machine Interface*)

Human Machine Interface (HMI) adalah sistem yang menghubungkan antara manusia dan teknologi mesin. HMI dapat berupa pengendali dan visualisasi status, baik dengan manual maupun melalui visualisasi komputer yang bersifat *realtime*. Sistem HMI biasanya bekerja secara online dan *realtime* dengan membaca data yang dikirimkan melalui I/O *port* yang digunakan oleh sistem *controller*-nya. *Port* yang biasanya digunakan untuk *controller* dan akan dibaca oleh HMI antara lain adalah *port com*, *port USB*, *port RS232* dan ada pula yang menggunakan *port serial*. Berikut ini merupakan fungsi dari HMI yaitu:

1. Memberikan informasi *plant* yang *up-to-date* kepada operator melalui *graphical user interface*.
2. Menerjemahkan instruksi operator ke mesin.
3. Memonitor keadaan yang ada di *plant*.
4. Mengatur nilai pada parameter yang ada di *plant*.
5. Mengambil tindakan yang sesuai dengan keadaan yang terjadi.

6. Memunculkan tanda peringatan dengan menggunakan alarm jika terjadi sesuatu yang tidak normal.
7. Menampilkan pola data kejadian yang ada di plant baik secara *realtime* maupun *historical* (*Trending history* atau *real time*).

2.11 Power Supply

Sistem *power supply* merupakan faktor yang paling penting dalam suatu sistem, baik yang bersifat analog maupun digital. Karena suatu sistem tidak akan berfungsi atau berjalan dengan baik tanpa mendapat sumber tegangan dan bisa dikatakan sebagai suatu rangkaian yang menyediakan daya. Arus yang dikeluarkan *power supply* bersifat searah dan tidak lagi bolak-balik, tegangan yang dihasilkan juga kecil hanya beberapa volt saja, beda dengan tegangan listrik PLN yaitu 220V. Bagian-bagian yang terdapat pada rangkaian *power supply* adalah sebagai berikut:

1. Step Down

Power supply menerima *input* dari jala-jala PLN sebesar 220V. Tegangan AC tersebut masuk ke *input* transformator, bagian primer trafo berfungsi menurunkan daya listrik dan tegangan yang ada bersifat bolak-balik atau *Alternating Current* (AC) dan belum rata.

2. Rectifier

Dengan menggunakan *dioda* silikon, maka tegangan AC akan disearahkan atau diubah menjadi tegangan DC, tetapi tegangan yang dihasilkan belum rata.

3. Filter

Tegangan yang belum rata, diratakan oleh *tapis* perata berupa kapasitor bipolar atau *electrolit condensator* (Elco), sehingga dihasilkan tegangan DC yang rata.

4. Stabilisator atau regulator

Tegangan yang melewati kapasitor tidaklah benar-benar rata atau stabil, dapat lebih tinggi dari *input* sekunder trafo ataupun dapat lebih rendah, sehingga diperlukan rangkaian stabilisator atau regulator untuk mengatasinya, sehingga keluaran yang dihasilkan benar-benar sesuai dengan yang diharapkan atau sesuai dengan *input* sekunder trafo. Komponen yang digunakan dapat berupa *diode zener*, transistor, atau IC.

