

BAB II

LANDASAN TEORI

2.1. Perangkat Keras

Dalam bab ini penulis akan menjelaskan tentang teori – teori yang berhubungan dengan alat yang dibuat, meliputi perangkat keras dan perangkat lunak.

Berikut adalah komponen - komponen perangkat keras untuk membuat alat Pagar Otomatis, yang berisikan seperti Arduino UNO sebagai otak dari sistem kerja semua komponen, sensor untuk mendeteksi objek/benda, motor Stepper sebagai penggerak pagar otomatis, LCD untuk menampilkan notifikasi akses dari Module RFID Kemudian Baterai sebagai sumber tenaga.

2.1.1 Teori IC Digital

Menurut (Maulana, 2017) mengatakan bahwa, “*Integrated Circuit (IC)* adalah Sirkuit terintegrasi atau yang biasa juga disebut sebagai IC merupakan komponen elektronika yang terbuat dari kumpulan puluhan, ratusan, hingga ribuan transistor, resistor, dioda dan komponen elektronika lainnya”.

IC mempunyai fungsi dan tugas khusus yang telah ditentukan oleh pembuatnya, tugas khusus ini ditentukan dengan bagaimana cara merangkai komponen yang ada karena pada dasarnya adalah sebuah rangkaian tertentu dengan fungsi tertentu pula. *IC* Digital beroperasi pada tegangan 0 volt (*low*) dan 5 volt (*high*). *IC* digital tersusun dari beberapa rangkaian logika *AND*, *OR*, *NOT*, *NAND*, *NOR*, dan *XOR*.

2.1.2. Sumber Tegangan

Menurut (Sitohang et al., 2018) memberikan batasan bahwa, “Catu daya (*Power Supply*) adalah sebuah perangkat yang memasok listrik energi untuk satu atau lebih beban listrik”.

Catu daya menjadi bagian yang penting dalam elektronika yang berfungsi sebagai sumber tenaga listrik misalnya pada baterai atau accu. Pada dasarnya power supply ini mempunyai konstruksi rangkaian yang hampir sama yaitu terdiri dari trafo, penyearah, dan penghalus tegangan. Istilah ini paling sering diterapkan ke perangkat yang mengubah satu bentuk energi listrik yang lain, meskipun juga dapat merujuk ke perangkat yang mengkonversi bentuk energi lain (misalnya, mekanik, kimia, solar) menjadi energi listrik.

Secara umum prinsip rangkaian catu daya terdiri atas komponen utama yaitu ; transformator, dioda dan kondensator. Dalam pembuatan rangkaian catu daya selain menggunakan komponen utama juga diperlukan komponen pendukung agar rangkaian berfungsi dengan baik Ada dua sumber catu daya yaitu sumber AC dan sumber DC. Sumber AC yaitu sumber tegangan bolak – balik, sedangkan sumber tegangan DC merupakan sumber tegangan searah.

Dibawah ini adalah jenih-jenis catu daya:

1. Sumber Arus Bolak Balik (*Alternating Current/AC*)

Arus listrik bolak balik adalah arus listrik dengan besar dan arah yang berubah-ubah secara bolak balik. Arus AC mengalir bolak balik dari potensial tinggi (+) ke potensial rendah (-) dan dari potensial rendah (-) ke potensial tinggi (+), Gelombang listrik pada arus AC berbentuk *sinusoidal*, gelombang segi empat atau gelombang segitiga, contoh penggunaan arus listrik bolak-balik adalah pada

jaringan PLN dan generator AC, jika menggunakan tegangan listrik PLN, besar arusnya berkisar antara 110 *volt* hingga 220 *volt* dengan frekuensi 50 *Hertz*.

2. Sumber Arus Searah (*DirectCurrent/ DC*)

Arus listrik searah adalah arus listrik yang bernilai konstan dan mengalir dari potensial tinggi (+) ke potensial rendah (-). Besar arus listrik yang sering ditemukan berkisar antara 1,5 *volt* hingga 24 *volt*. Arus listrik searah biasa digunakan pada baterai, dinamo arus searah atau aki, sumber tegangan ini tidak mengalami perubahan terhadap waktu.



Sumber : <http://corelita.com/pengertian-adaptor-atau-catu-daya/>

Gambar II.1.

Catu Daya

2.1.3. Komponen Elektronika

Menurut (Maulana, 2017) disimpulkan bahwa, “Rangkaian elektronika, komponen-komponen elektronika dibagi dalam jenis komponen pasif dan komponen aktif”

1. *Light Emitting Diode (LED)*

Menurut (Suhardi, 2014) LED adalah semikonduktor yang dapat mengubah energi listrik lebih banyak menjadi cahaya, merupakan perangkat keras dan padat (solid-state component) sehingga lebih unggul dalam ketahanan (durability).



Sumber : https://www.123rf.com/photo_19706682_set-of-color-3-mm-led-diodes-isolated-on-white-background.html

Gambar II.2

LED

1. **Resistor**

Menurut (Listiyarini, 2018) memberikan batasan bahwa “Resistor disebut juga dengan tahanan atau hambatan, merupakan komponen pasif elektronika yang berfungsi untuk membatasi arus listrik yang mengalir pada suatu rangkaian listrik”.



Sumber : <http://www.instructables.com/lesson/Resistors/>

Gambar II.3

Resistor

2. Kapasitor

Menurut (Maulana, 2017) “Kapasitor merupakan komponen pasif yang sering digunakan pada sistem yang berfungsi sebagai filter dan penyimpan energi listrik”.



Sumber : <http://www.uniksharianja.com/2016/02/cara-menentukan-nilai-kapasitas-kapasitor.html>

Gambar II.4.

Kapasitor

3. Dioda

Menurut (Maulana, 2017) memberikan batasan bahwa, “Dioda adalah komponen elektronika yang membuat arus listrik mengalir hanya dalam satu arah, sehingga biasa disebut juga sebagai penyearah”.



Sumber : <http://lapantech.com/Dioda-1n4002>

Gambar II.5

Dioda

4. Konektor

Menurut (Maulana dan Rachmat Adi Purnama, 2017) memberikan batasan bahwa, “Konektor yang digunakan dalam rangkaian elektronika, untuk memudahkan melepas pasang rangkaian konektor ini memiliki lubang pin beragam, dan disesuaikan dengan kebutuhan”.

Header atau biasa dikenal dengan nama pin *head* berguna sebagai soket tempat menghubungkan kabel-kabel konektor, ada dua jenis pin14 *header* yaitu pin *header male* dan pin *header female*, walaupun berbeda namun memiliki fungsi yang sama yaitu sebagai konektor.



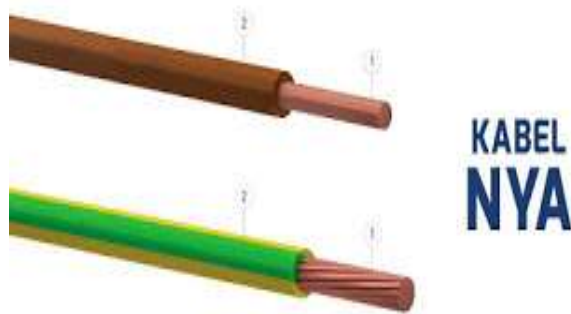
Sumber : <https://electronics.stackexchange.com/questions/37783/how-can-i-create-a-female-jumper-wire-connector>

Gambar II.6.

Konektor

5. Kabel Jumper

Menurut (Maulana dan Rachmat Adi Purnama, 2017) memberikan batasan bahwa, “Kabel Jumper digunakan untuk menyalurkan energi listrik. Sebuah kabel listrik terdiri dari isolator dan konduktor isolator adalah bahan pembungkus kabel yang biasa terbuat dari plastik atau karet, sedangkan konduktor terbuat dari tembaga”.



Sumber : https://klinikkonstruksi.jogjaprovo.go.id/artikel_detail.php?id=99

Gambar II. 7

Kabel Jumper

6. *Push Button Switch*

Menurut (Sokop, Jendri Steven, Dringhuzen J. Mamahit, 2016) memberikan batasan bahwa, “Push button switch (saklar tombol tekan) adalah perangkat / saklar sederhana yang berfungsi untuk menghubungkan atau memutuskan aliran arus listrik dengan sistem kerja tekan unlock (tidak mengunci)”.

Sistem kerja unlock disini berarti saklar akan bekerja sebagai device penghubung atau pemutus aliran arus listrik saat tombol ditekan, dan saat tombol tidak ditekan (dilepas), maka saklar akan kembali pada kondisi normal.



Sumber : <https://www.sparkfun.com/products/97>

Gambar II. 8

Push Button Switch

2.1.4. Sensor Ultrasonik

Menurut (Arasada, 2017) memberikan batasan bahwa, “Sensor Ultrasonik adalah sensor yang bekerja berdasarkan prinsip pantulan gelombang suara dan digunakan untuk mendeteksi keberadaan suatu objek atau benda tertentu didepan frekuensi kerja pada daerah diatas gelombang suara dari 20 kHz hingga 2 MHz”.

1. Sensor Ultrasonik HCSRFB-04

Prinsip kerja sensor ini adalah transmitter mengirimkan sebuah gelombang ultrasonik lalu diukur dengan waktu yang dibutuhkan hingga datangnya pantulan dari objek. Lamanya waktu ini sebanding dengan Aplikasi Sensor Ultrasonik Untuk Deteksi Posisi Jarak Pada Ruang Menggunakan Arduino Uno. Dua kali jarak sensor dengan objek, sehingga jarak sensor dengan objek dapat ditentukan persamaan 1: $s = v \times t$ 2 Keterangan: s = jarak (meter) v = kecepatan suara (344 m/detik) t = waktu tempuh (detik) Hcsrf-04 dapat mengukur jarak dalam rentang antara 3cm–3m dengan output panjang pulsa yang sebanding dengan jarak objek. Sensor ini hanya memerlukan 2 pin I/O untuk berkomunikasi dengan mikrokontroler, yaitu TRIGGER dan ECHO. Untuk mengaktifkan HCSRFB-04 mikrokontroler mengirimkan pulsa positif melalui pin TRIGGER minimal 10 μ s, selanjutnya HCSRFB-04 mengirimkan pulsa positif melalui pin ECHO selama 100 μ s hingga 18 ms, yang sebanding dengan jarak objek. Spesifikasi dari sensor ultrasonik HCSRFB-04 adalah sebagai berikut: a. Dimensi: 24mm (P) x 20mm (L) x 17mm (T). b. Konsumsi Arus: 30 mA (rata-rata), 50 mA (max). c. Jangkauan: 3 cm–3 m. d. Sensitifitas: Mampu mendeteksi objek dengan diameter 3 cm pada jarak > 1m.



Sumber ; <https://www.tokopedia.com/arduinomakassar/sensor-ultrasonic-hc-sr04>

Gambar II.9

. Sensor ultrasonik HC-SR04

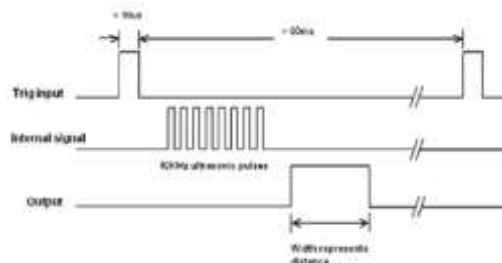
Tabel II. 1

Pin-pin pada HC-SR04

Pin	Keterangan
Pin 1	Vcc (dihubungkan ke tegangan +5V)
Pin 2	Trig (untuk mengirimkan gelombang suara)
Pin 3	Echo (untuk menerima pantulan gelombang suara)
Pin 4	Gnd (dihubungkan ke <i>ground</i>)

2. Prinsip Kerja Sensor Ultrasonik HCSR04

Prinsip kerja HCSR04 adalah transmitter memancarkan seberkas sinyal ultrasonik (20 KHz) yang berbentuk pulsa, kemudian jika didepan HCSR04 ada objek padat maka receiver akan menerima pantulan sinyal ultrasonik tersebut Receiver akan membaca lebar pulsa (dalam bentuk PWM) yang dipantulkan objek dan selisih waktu pemancaran.



Sumber :<https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>

Gambar II.10

Timing diagram HC-SR04

3. *Radio Frequency Identification (RFID)*

Menurut (Mubarok, Sofyan, Rismayadi, & Najiyah, 2018) memberikan batasan bahwa, “RFID adalah teknologi identifikasi berbasis gelombang radio. Teknologi ini mampu mengidentifikasi berbagai objek secara simultan tanpa diperlukan kontak langsung (atau dalam jarak pendek)”.

1. *Mifare RC522*

RFID Reader Module adalah sebuah modul berbasis IC *Philips MFRC522* yang dapat membaca RFID dengan penggunaan yang mudah dan harga yang murah, karena modul ini sudah berisi komponen-komponen yang diperlukan oleh MFRC522 untuk dapat bekerja. Modul ini dapat digunakan langsung oleh MCU dengan menggunakan interface SPI, dengan suplai tegangan sebesar 3,3 volt. MFRC522 merupakan produk dari NXP yang menggunakan *fully integrated 13.56MHz non-contact communication card chip* untuk melakukan pembacaan maupun penulisan. MFRC522 support dengan semua varian MIFARE Mini, MIFARE 1K, MIFARE 4K, MIFARE *Ultralight*, MIFARE DESFire EV1 and MIFARE Plus RF *identification Protocols*.



Sumber : <https://www.indo-ware.com/produk-2865-rfid-reader-writer-module-mifare-rc522-1356-mhz-.html>

Gambar II.11.

Mifare RC522

Tabel II. 2
Pin Modul Reader RFID MFRC522

NO	Pin
1.	SDA
2.	SCK
3.	MOSI
4.	MISO
5.	IRQ
6.	GND
7.	RST
8.	3.3V

2. Tag RFID

Tag ini bekerja saat antena mendapatkan sinyal dari *reader* RFID dan sinyal tersebut akan dipantulkan lagi, sinyal pantul ini biasanya sudah ditambahkan dengan data yang dimiliki tag tersebut. RFID tag ukurannya dapat berbeda-beda, pada umumnya kecil. Tag yang ditanamkan ke dalam metal, kunci ,dan sebagainya.



Sumber : <https://www.indiamart.com/proddetail/rfid-tag-card-17920917273.html>

Gambar II.12

Tag RFID

2.1.5. Output

Sebuah alat tidak dapat bekerja apabila tidak ada *output*. Begitu juga dengan alat yang kami buat yaitu alat *prototype* pagar otomatis. Berikut adalah komponen penggerak (*output*) yang kami gunakan dalam pembuatan alat pagar otomatis yaitu *motor stepper*, *motor driver*, I2C, dan LCD. Uraian dari masing – masing hal tersebut akan dijelaskan sebagai berikut.

1. *Motor Stepper (28BYJ-48)*

Motor Stepper yang digunakan pada alat ini yaitu *motor stepper* 28BYJ48.

Motor stepper ini merupakan jenis motor stepper Unipolar. Motor ini memiliki rasio gear sebesar 64, dan putaran per stepnya sebesar 5.625° jadi motor ini memiliki step sebanyak 4096. Untuk menghitung jumlah step dalam satu putarannya dapat menggunakan rumus dibawah ini :

steps = Jumlah step per satu Putaran * Rasio Gear

steps = $(360^\circ/5.625^\circ) * 64$ steps = $64 * 64$ steps =

4096



Sumber : <https://www.instructables.com/id/Supercharge-and-Convert-a-Unipolar-Stepper-to-Bipo/>

Gambar II. 13

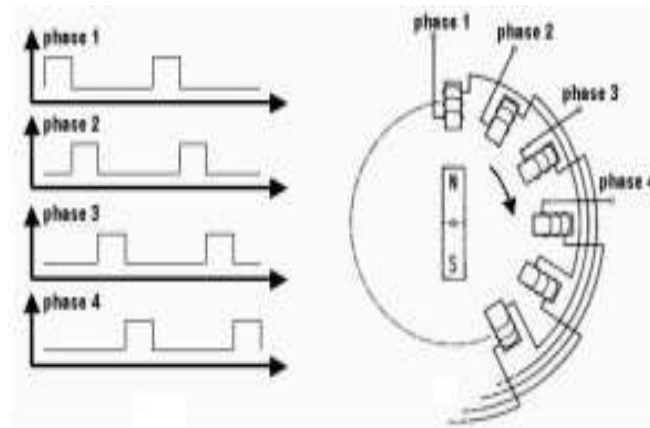
Motor Stepper

Tabel II. 3
Spesifikasi Motor Stepper 28BYJ-48

Related Voltage	5 VDC
Number of Phase	4
Speed Variation Ratio	6-20 Volt
Stride Angle	5.625°/64
Frequency	100Hz
DC Resistance	50Ω±7%(25°C)
Idle In-traction Frequency	> 600Hz
Idle Out-traction Frequency	> 1000Hz
In-traction Torque	>34.3mN.m(120Hz
Self-positioning Torque	>34.3mN.m
Friction Torque	600-1200 gf.cm
Pull in Torque	300 gf.cm
Insulated Resistance	>10MΩ(500V)
Insulated Elecrticity	600VAC/1mA/1s
Insulated Grade	A
Rise in Temperature	<40K(120Hz)
Noise	<35dB(120Hz,No load,10cm)
Model	28BYJ-48 – 5V

a. Prinsip Kerja Motor Stepper

Prinsip kerja motor stepper adalah mengubah pulsa-pulsa input menjadi gerakan mekanis diskrit. Oleh karena itu untuk menggerakkan motor stepper diperlukan pengendali motor stepper yang membangkitkan pulsa-pulsa periodik. Berikut ini adalah ilustrasi struktur *motor stepper* sederhana dan pulasa yang dibutuhkan untuk menggerakkannya



Sumber : <http://zoniaelektro.net/motor-stepper/>

Gambar II. 14

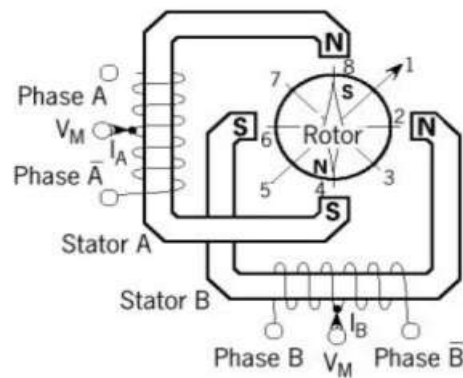
Prinsip Kerja *Motor Stepper*

Gambar diatas memberikan ilustrasi dari pulsa keluaran pengendali motor stepper dan penerpan pulsa tersebut pada motor stepper untuk menghasilkan arah putaran yang bersesuaian dengan pulsa kendali.

b. Jenis – Jenis *Motor Stepper*

1) *Motor Stepper Unipolar*

Rangkaian pengendali motor stepper unipolar lebih mudah dirancang karena hanya memerlukan satu switch / transistor setiap lilitannya. Untuk menjalankan dan menghentikan motor ini cukup dengan menerapkan pulsa digital yang hanya terdiri atas tegangan positif dan nol (ground) pada salah satu terminal lilitan (wound) motor sementara terminal lainnya dicatu dengan tegangan positif konstan (VM) pada bagian tengah (center tap) dari lilitan seperti pada gambar berikut.



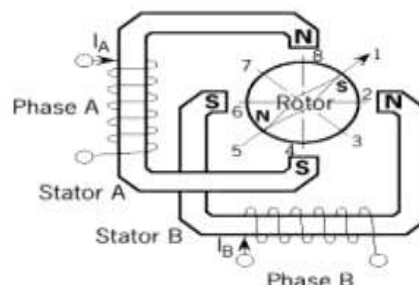
Sumber : <http://zoniaelektro.net/motor-stepper/>

Gambar II. 15

Motor stepper dengan lilitan unipolar

2) Motor Stepper Bipolar

Untuk *motor stepper* dengan lilitan bipolar, diperlukan sinyal pulsa yang berubah-ubah dari positif ke negatif dan sebaliknya. Jadi pada setiap terminal lilitan (A & B) harus dihubungkan dengan sinyal yang mengayun dari positif ke negatif dan sebaliknya. Karena itu dibutuhkan rangkaian pengendali yang agak lebih kompleks daripada rangkaian pengendali untuk motor unipolar. *Motor stepper bipolar* memiliki keunggulan dibandingkan dengan motor stepper unipolar dalam hal torsi yang lebih besar untuk ukuran yang sama.



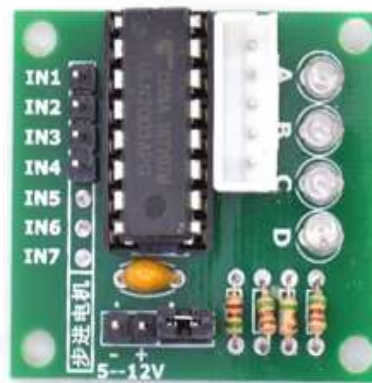
Sumber: <http://zoniaelektro.net/motor-stepper/>

Gambar II. 16

Motor stepper dengan lilitan bipolar

2. *Motor Driver ULN2003*

Motor Driver ULN2003 merupakan perangkat yang mempermudah mengontrol Motor Stepper 28BYJ-48 dari mikrokontroler, seperti Arduino. Pada motor driver ULN2003 ini terdapat 5 *socket* kabel untuk menyambungkan kabel dari motor stepper 28BYJ-48 dan 4 LED untuk menunjukkan coil yang sedang diberi tegangan. Untuk tegangannya, dianjurkan menggunakan power supply 5-12Volt 1Amp. Dan pada motor driver ini juga terdapat 4 input kontrol yang harus dihubungkan ke empat Arduino pin digital.



Sumber: <https://www.barerobotics.com/product/uln2003-unipolar-stepper-motor-driver-12v-05a>

Gambar II. 17

Motor Driver ULN2003

3. I2C (*Inter Integrated Circuit*)

Menurut (Sari, Kartika, Cucu Suhery, 2015) memberi batasan bahwa, “I2C merupakan singkatan dari *Inter Integrated Circuit*, yaitu sebuah protokol untuk komunikasi serial antar IC dan sering disebut juga *Two Wire Interface* (TWI). Pada penelitian ini, komunikasi I2C digunakan untuk komunikasi antara mikrokontroler dan *RTC*”.

Komunikasi I2C merupakan standar komunikasi dengan menggunakan 2 kabel dengan kecepatan yang cukup tinggi sehingga antara IC yang satu dengan yang lainnya dapat berkomunikasi dengan baik. Pada penelitian ini, komunikasi I2C digunakan untuk menghubungkan antara RTC dengan Arduino.



Sumber : <http://www.dx.com/p/lcd1602-adapter-board-w-iic-i2c-interface-black-works-with-official-arduino-boards-216865>

Gambar II.18

I2C

4. *LCD (Liquid Crystal Display)*

Menurut (Dinata dan Wahri Sunanda, 2015) memberikan batasan bahwa, “LCD berfungsi untuk menampilkan suatu nilai hasil sensor, menampilkan teks, atau menampilkan menu pada aplikasi mikrokontroler”.

Modul LCD matrix tersedia dengan konfigurasi 16 karakter dan 2 baris dengan setiap karakternya dibentuk oleh baris pixel.

Adapun fitur yang disajikan dalam LCD ini adalah:

- a. Terdiri dari 16 karakter dan 2 baris.
- b. Mempunyai 192 karakter tersimpan.
- c. Terdapat karakter generator terprogram.
- d. Dapat dialamati dengan mode 4-bit dan 8-bit.
- e. Dilengkapi dengan *back light*.



Sumber : <https://www.indiamart.com/proddetail/adraxx-hd44780-16-x-2-lcd-modulediy-general-microcontroller-16393243033.html>

Gambar II.19

LCD

5. *Buzzer*

Menurut (Mubarak et al., 2018) memberikan batasan bahwa, “Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara.”.

Prinsip kerja buzzer hampir sama dengan *loud speaker*, jadi buzzer juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara.



Sumber: <https://www.indiamart.com/proddetail/horn-buzzer-15460445088.html>

Gambar II. 20

Buzzer

2.1.6. Mikrokontroler

Menurut (Sujarwata, 2018) mengatakan bahwa, “Mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran (*I/O*) serta pengendali (kontrol) dengan suatu program yang dapat ditulis dan dihapus dengan cara khusus”.

Mikrokontroler berfungsi sebagai pengontrol rangkaian elektronika dan pada umumnya dapat menyimpan program yang pada dasarnya menggunakan bahasa *assembler*. Saat ini mikrokontroler dapat diprogram dengan menggunakan bahasa tingkat tinggi seperti *BASIC*, *PASCAL* atau *C*. Agar semua mikrokontroler dapat berfungsi, maka mikrokontroler tersebut memerlukan komponen eksternal yang kemudian disebut dengan *system minimum*. Untuk membuat sistem minimal paling tidak dibutuhkan sistem *clock* dan *reset*, walaupun pada beberapa mikrokontroler sudah menyediakan sistem *clock* internal, sehingga tanpa rangkaian eksternal pun mikrokontroler sudah beroperasi. Mikrokontroler berbeda dari mikroprosesor serba guna yang digunakan dalam sebuah *personal computer (PC)*, karena sebuah mikrokontroler pada umumnya telah berisi komponen pendukung sistem minimal mikroprosesor, yakni memori dan antarmuka *I/O*. Mikrokontroler cenderung beroperasi pada manipulasi bit, sedangkan mikroprosesor cenderung beroperasi pada operasi *byte* (8bit).

Pada alat pagar otomatis ini menggunakan mikrokontroler ATmega328P, dan implementasi yang digunakan dari ATmega328P adalah *platform* pengembangan Arduino yaitu model Arduino Uno.

Sedangkan Arduino menurut (Ahyadi, 2018) mengatakan bahwa “Arduino adalah suatu *open-source platform* elektronik yang berbasis kemudahan penggunaan (*easy to use*) baik *hardware* maupun *software*”.

1. Arduino Uno

Menurut (Kadir, 2017) mengatakan bahwa, “Arduino Uno adalah papan yang mengandung sebuah mikrokontroler buatan *Atmel* (Atmega328p) yang menjadi pusat pengendali perangkat keras dan sejumlah pin untuk kepentingan operasi masukan (*input*) dan keluaran (*output*)”.

Hardware mikrokontroler Arduino diprogram dengan menggunakan bahasa pemrograman *wiring-based* yang berbasiskan *syntax* dan *library*. Pemrograman *wiring-based* ini tidak jauh berbeda dengan C/C++, tetapi dengan beberapa penyederhanaan dan modifikasi. Untuk memudahkan dalam pengembangan aplikasinya, mikrokontroler Arduino juga menggunakan *Integrated Development Environment (IDE)* berbasis *processing*. Mikrokontroler Arduino dapat dipasangkan dengan bermacam-macam sensor. Adapun sensor yang dapat dipasangkan pada Arduino seperti sensor gerak, ultrasonik, panas, suara, asap, gas, dan lain-lain.



Sumber : Akhiruddin (2017:36)

Gambar II.21

Arduino Uno

Konfigurasi yang terdapat pada Arduino Uno menurut (Sokop, Jendri Steven, Dringhuzen J. Mamahit, 2016) adalah:

- a. Mikrokontroler : ATmega328
- b. Tegangan pengoperasian : 5V
- c. Tegangan *input* yang disarankan: 7-12V.
- d. Batas tegangan *input* : 6-20V.
- e. Jumlah pin *I/O* digital : 14.
- f. Jumlah pin *input* analog : 6.
- g. Arus *DC* tiap pin *I/O* : 40 mA.
- h. Arus *DC* untuk pin 3.3V : 50 mA.
- i. Memori : 32 KB (ATmega328).

sekitar 0.5 KB digunakan oleh *bootloader*.

- j. SRAM : 2 KB (ATmega328).
- k. EEPROM : 1 KB (ATmega328).
- l. Clock Speed : 16 MHz.

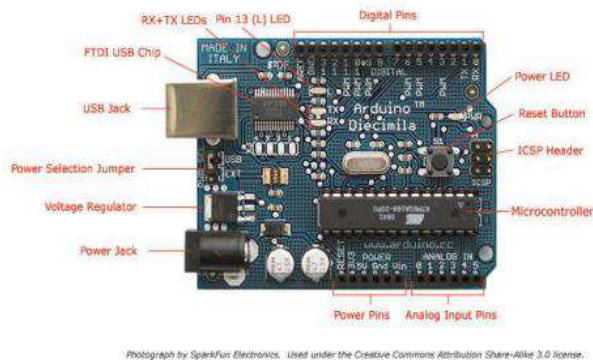


Sumber: <http://www.labelektronika.com/2017/02/arduino-uno-mikrokontroler-atmega-328.html>

Gambar II.22

Kaki Pin Atmega328

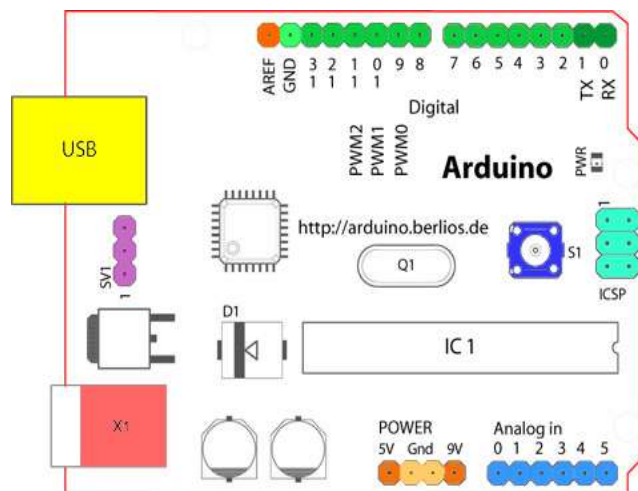
2. Berikut ini bagian-bagian Arduino Uno yang terdapat pada gambar:



Sumber : Dinata dan Wahri Sunanda (2015:85)

Gambar II.23

Bagian-bagian Arduino Uno



Sumber : <https://www.arduino.cc/en/reference/board>

Gambar II.24

Skema Bagian-bagian Arduino Uno

- a. *USB*

USB digunakan untuk mengupload program dari komputer ke dalam *Board* Arduino dan digunakan untuk memberikan catu daya ke Papan Arduino menggunakan kabel *USB* dari komputer.

b. Kristal (*quartz crystal oscillator*)

Kristal merupakan jantung dari *Board* Arduino itu sendiri, dikarenakan kristal menghasilkan detak-detak yang dikirim kepada mikrokontroler agar melakukan sebuah operasi untuk setiap detaknya, yang biasanya berdetak 16 juta kali per detik (16MHz).

c. Tombol *Reset*

Tombol *reset* digunakan untuk mereset *Board* Arduino sehingga program akan memulai dari awal kembali. Tombol ini tidak untuk menghapus program atau mengosongkan mikrokontroler.

d. Pin Analog *Input* (0 – 5)

Pin analog digunakan untuk membaca tegangan yang dihasilkan oleh sensor analog. Nilai pin *output* analog dapat diprogram dari 0 – 255 dan dapat membaca nilai analog dari 0 – 1023, yang keduanya mewakili nilai tegangan 0-5 Volt.

e. Pin Digital *I/O* (0 – 13)

14 pin digital ini berfungsi sebagai *input* atau *output* dan dapat diatur oleh program. Khusus untuk pin 3, 5, 6, 9, 10, dan 11 dapat juga berfungsi sebagai pin analog *output* yang tegangan *output*nya dapat diatur.

f. Eksternal *Power Supply*

Sumber tenaga eksternal adalah penyuplai tegangan DC sebesar 9–12 Volt.

g. *In Circuit Serial Programming (ICSP)*

Port *ICSP* digunakan untuk memprogram mikrokontroler tanpa melalui *bootloader*. *ICSP* menggunakan jalur *Serial Peripheral Interface (SPI)* untuk transfer data. Namun pada umumnya para pengguna tidak melakukan hal ini.

h. *IC* – mikrokontroler atmega

IC merupakan komponen utama dari *Board* Arduino yang didalamnya terdapat *CPU*, *ROM*, dan *RAM*.

i. Pin *Power*

Pin *power* yang terdapat pada Arduino Uno adalah sebagai berikut:

- 1) *Vin* = pin yang digunakan jika ingin memberikan *power* langsung ke *board* Arduino dengan tegangan 7V – 12V.
- 2) *5V* = pin *output* 5V yang telah diatur oleh regulator *board* Arduino. *Board* dapat diaktifkan dengan daya dari *Power Supply* (7 - 12V), *USB* (5V), atau pin *Vin board* (7-12V). Jika tegangan di *input* melalui pin 5V atau 3.3V secara langsung tanpa melewati regulator, maka dapat merusak *board* Arduino.
- 3) 3.3V = dihasilkan oleh regulator *on board*. Menyediakan arus maksimum 50 *mA*.
- 4) GND = Pin *Ground*.
- 5) IOREF = memberikan tegangan referensi ketika mikrokontroler beroperasi. Biasa digunakan pada *board shield* untuk dapat memilih sumber daya yang tepat agar dapat bekerja dengan 5V atau 3.3V.

2.2. Perangkat Lunak

Menurut (Swara, Kom, & Pebriadi, 2016) memberi batasan bahwa, “Perangkat Lunak merupakan seluruh perintah yang digunakan untuk memproses informasi. Perangkat lunak dapat berupa program maupun prosedur yang didalamnya merupakan kumpulan perintah yang dimengerti oleh komputer sedangkan prosedur adalah perintah yang dibutuhkan oleh pengguna dalam memproses informasi”.

Perangkat lunak atau *software* yang digunakan dalam pembuatan alat pagar otomatis menggunakan sensor sonar, rfid mfrc522 ini adalah *software proteus*, *Flowchart*, *Software Arduino IDE (Integrated Development Environment)* yang menggunakan bahasa C.

1. *Flowchart*

Menurut (Nuraini, 2015) menyimpulkan bahwa:

“*Flowchart* dapat diartikan sebagai suatu alat atau sarana yang menunjukkan langkah-langkah yang harus dilaksanakan dalam menyelesaikan suatu permasalahan untuk komputasi dengan cara mengekspresikannya ke dalam serangkaian simbol-simbol grafis khusus, *flowchart* sangat berguna khususnya untuk menjelaskan urutan-urutan proses yang pelaksanaannya mempunyai banyak pilihan atau percabangan”.

Bentuk diagram alur (*flowchart*) yang sering digunakan dalam proses pembuatan suatu program komputer adalah sebagai berikut:

a. Program *Flowchart*

Simbol-simbol yang menggambarkan suatu urutan prosedur secara rinci dan detail antara instruksi yang satu dengan yang lainnya di dalam suatu program komputer yang bersifat *logic*.

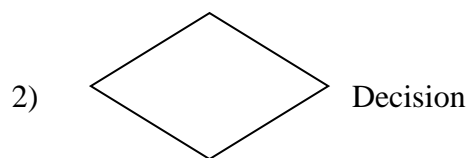
b. Sistem *Flowchart*

Simbol-simbol yang menggambarkan urutan prosedur secara *detail* di dalam suatu sistem komputer, dan bersifat fisik.

Simbol-simbol yang digunakan adalah sebagai berikut.



Menyatakan awal dan akhir dari suatu kegiatan.



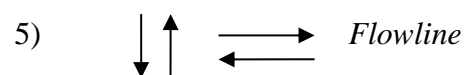
Simbol ini menunjukan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban ya/tidak.



Simbol proses menggambarkan proses yang ada dalam program.



Simbol yang menggambarkan masukan dan keluaran dalam program

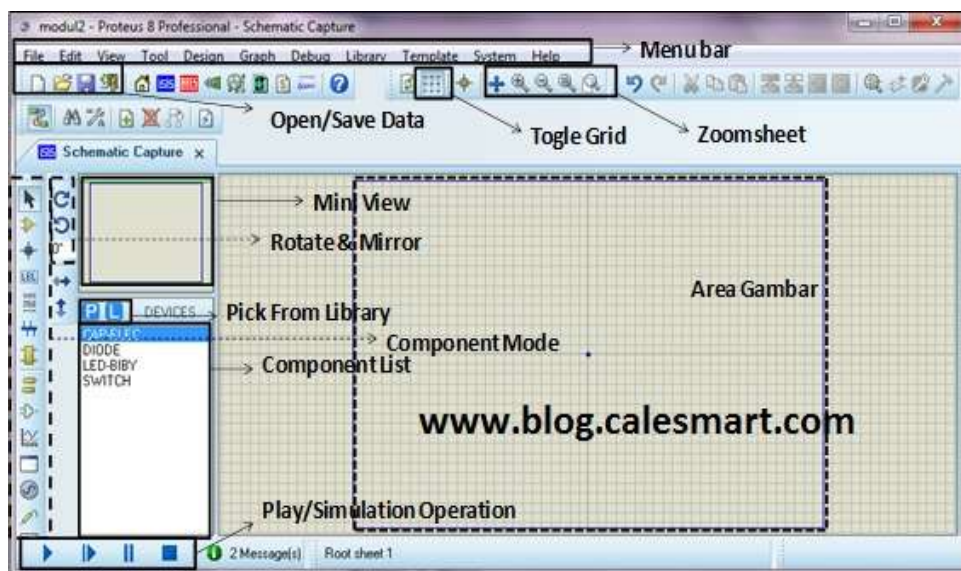


Simbol yang digunakan untuk menggambarkan hubungan proses dari satu proses ke proses lainnya.

2. *Software Proteus*

Menurut (Falani, 2015) memberikan batasan bahwa, “*Proteus* adalah sebuah software untuk mendesain *PCB* yang juga dilengkapi dengan simulasi pspice pada level skematik sebelum rangkaian skematik diupgrade ke *PCB* sehingga sebelum *PCB*nya di cetak, kita akan tahu apakah *PCB* yang akan kita cetak sudah benar atau tidak”.

Proteus mengkombinasikan program *ISIS* untuk membuat skematik desain rangkaian dengan program *ARES* untuk membuat layout *PCB* dari skematik yang kita buat. Software ini bagus digunakan untuk desain rangkaian mikrokontroller. Proteus juga bagus untuk belajar elektronika seperti dasar-dasar elektronika sampai pada aplikasi mikrokontroller. Software ini jika di install menyediakan banyak contoh aplikasi desain yang disertakan sehingga kita bisa belajar dari contoh-contoh yang sudah ada.



Sumber: https://calesmart.com/artikel/Pengenalan-simulasi-elektronika-dengan-Software-Proteus-8-Profesional_132.html

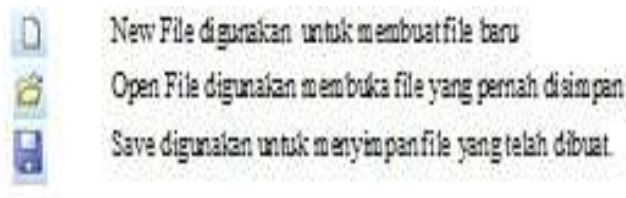
Gambar II. 25

Tampilan Skematik Proteus

Keterangan *Menu Bar* pada *Proteus* :

- a. Menu Bar Merupakan List menu yang dapat digunakan dalam perancangan atau pengolahan rangkaian.

- b. *Open, save*



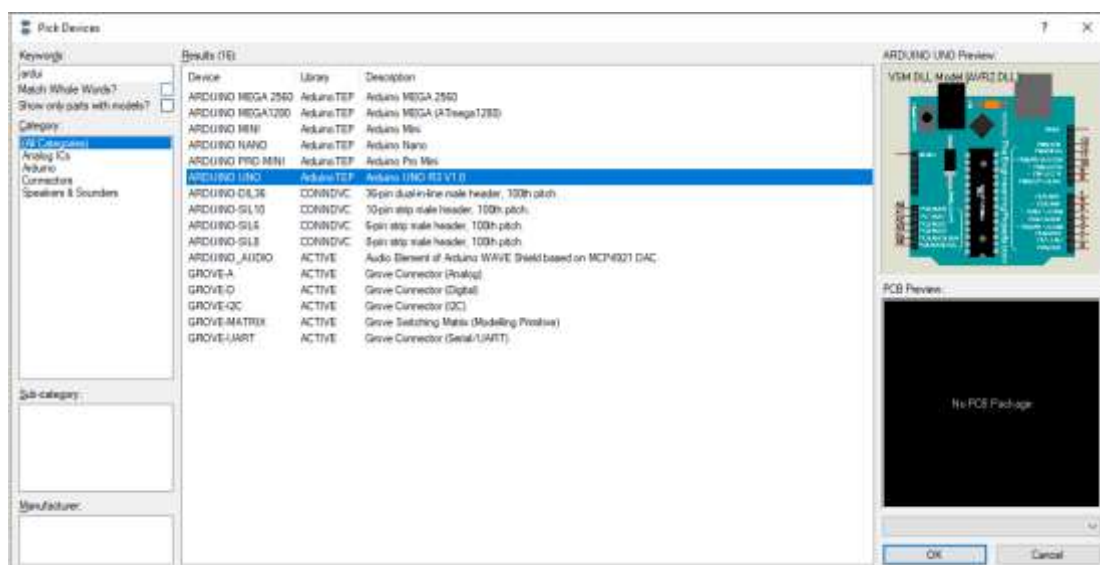
- c. *Toggle grid* digunakan untuk menampilkan bantuan titik-titik panduan pada area gambar.
- d. *Zoom sheet* untuk menentukan area tengah gambar dengan betumpu pada kursor, memperbesardan memperkecil gambar.
- e. Menu View Digunakan untuk menampilkan gambar dalam bentuk tampilan kecil seluruh area gambar.
- f. *Componen list* merupakan daftar komponen yang telah diambil dari library
- g. *Componen Mode*



- h. *Rotate* dan *Mirror* digunakan untuk merotasi obyek.
- i. *Play* dan *Simulation Operation* digunakan untuk menjalankan simulasi rangkaian yang telah dibuat.

Langkah – langkah pembuatan rangkaian simulasi di dalam proteus dapat diuraikan sebagai berikut :

- a. Jalankan Program Proteus
- b. Langkah selanjutnya adalah mengambil komponen yang dibutuhkan, dengan cara pilih mode komponen atau klik symbol P (pick from library) untuk Menuju library.
- c. Akan muncul window baru, selanjutnya mengambil komponen yang dibutuhkan dengan mengetikkan keyword pada kolom keyword.
- d. Menambil komponen yang dibutuhkan dilakukan dengan cara mengetikkan keyword komponen atau double klik pada komponen yang dipilih atau sampai muncul pada list komponen pada window utama.



Sumber : Dokumen Pribadi

Gambar II. 26

Mengambil Komponen

2.2.1. Bahasa Pemrograman

Menurut (Iswanto dan Nia Maharani Raharja, 2015) menyimpulkan bahwa:

“Bahasa C pertama kali digunakan di komputer *Digital Equipment Corporation* PDP-11 yang menggunakan sistem operasi *UNIX* C adalah bahasa yang standar, artinya suatu program yang ditulis dengan bahasa C tertentu akan dapat dikonversi dengan bahasa C yang lain dengan sedikit modifikasi, Standar bahasa C yang asli adalah standar dari *UNIX*”.

Berikut ini adalah fungsi-fungsi dari bahasa C :

1. Struktur

Struktur dalam penulisan program bahasa C mencakup tiga hal yaitu File *Header*, Fungsi Utama, Fungsi Lain. Pemrograman bahasa C biasa menggunakan fungsi utama *Main()* yang akan dijalankan pertama kali pada saat eksekusi program, meskipun didalam fungsi utama ini terdapat statement yang memanggil fungsi lain. Sedangkan didalam program *Arduino (Sketch)* memiliki dua fungsi yang diperlukan yang setiap fungsinya melampirkan blok-blok pernyataan.

```
void setup()
```

```
{
```

```
statements;
```

```
}
```

```
void loop()
```

```
{
```

```
statements;
```

```
}
```

a. *Void setup() { }*

Fungsi *setup()* akan dipanggil ketika *sketch* dijalankan pertama kali dan hanya akan berjalan sekali, yaitu setiap *power-up* atau *restart board* Arduino. Struktur fungsi ini untuk menginisialisasi mode pin, *variabel*, memulai penggunaan *library*, dan lain-lain.

b. *Void Loop() { }*

Setelah fungsi *setup()* selesai, fungsi *loop()* akan dijalankan secara terus menerus sampai catu daya (*power supply*) dilepas. Fungsi ini secara aktif mengontrol *board* Arduino baik secara *input* maupun *output*.

2. *Further Syntax*

- a. ; (titik koma) digunakan pada akhir setiap baris kode.
- b. {} (kurung kurawal) untuk mendefinisikan awal fungsi dan akhir fungsi.
- c. // (baris komentar) digunakan untuk memberi satu baris catatan dari arti kode-kode yang dituliskan.
- d. /**/ (blok komentar) digunakan untuk memberi dua atau lebih baris catatan dari arti kode-kode yang dituliskan.
- e. *#include* yaitu satu jenis pengarah praprosesor yang dipakai untuk membaca file yang dinamakan file judul (*header file*).

3. *Variabel*

Variabel adalah cara penamaan dan menyimpan nilai numerik yang nantinya digunakan oleh program. Sebuah *variabel* perlu dinyatakan dan diberikan secara opsional ke nilai yang perlu disimpan. *int inputVariable*

= 0; // menjelaskan sebuah *variabel* dan memberi nilai 0 *inputVariable* =
 analogRead(2); // menetapkan *variabel* ke nilai pin analog 2 *input*
Variable adalah *variabel* itu sendiri.

4. Tipe Data

- a. *Char* : Memiliki ukuran memori 1 *Byte* dengan jangkauan nilai -128 s/d 127 dan menyimpan 1 karakter menggunakan kode *ASCII*.
- b. *Byte* : Menyimpan nilai numerik 8-bit tanpa titik desimal dan memiliki jangkauan nilai 0-255.
- c. *Int* : Menyimpan bilangan tanpa titik desimal dan memiliki ukuran memori 2 *Byte*(16-bit) dengan jangkauan nilai -32,768 s/d 32,767.
- d. *Long* : Tipe data *extended* untuk bilangan *long integer* dan memiliki ukuran memori 4 *Byte* dengan jangkauan nilai -2,147,435,648 s/d 2,147,435,647.
- e. *Float* : Tipe data untuk angka yang memiliki titik desimal dan memiliki ukuran memori 4 *Byte* dengan jangkauan nilai -3.4028235E+38 s/d 3.4028235E+38.
- f. *Double* : Memiliki ukuran memori 8 *Byte* dengan jangkauan nilai - $1,7 \times 10^{(-308)}$ s/d $1,7 \times 10^{(+308)}$.
- g. *Boolean* : Digunakan untuk menyimpan nilai *TRUE* (benar) atau *FALSE* (salah) dan hanya menggunakan 1 bit dari *RAM*.

5. Array

Adalah tipe terstruktur yang terdiri dari sejumlah komponen-komponen yang mempunyai tipe yang sama. Banyaknya komponen dalam array ditunjuk oleh suatu indeks untuk membedakan antar variabel.

Format: TipeData NamaVariabel [*ukuran*]

- a. *Type* data, untuk menyatakan *type* data yang digunakan.
- b. Nama variabel, untuk menyatakan nama variabel yang digunakan.
- c. Ukuran, untuk menyatakan jumlah maksimum elemen array.

Contoh: *Float* Jumlah[8]

6. Operator Bahasa

Merupakan simbol atau kata yang digunakan dalam program untuk melakukan suatu operasi atau manipulasi, seperti menjumlahkan dua buah nilai, memberikan nilai suatu variabel, membandingkan dua buah nilai, dan lain-lain. Berikut ini operator-operator yang biasa digunakan dalam bahasa C, yaitu:

a. Operator Aritmatika

Operator ini digunakan untuk memanipulasi angka yang bekerja seperti matematika sederhana. Operator Aritmatika digolongkan dalam Operator *binary* dan operator *unary*.

Tabel II.4

Tabel Operator *Binary*

Operator	Keterangan	Contoh
+	Penjumlahan	4 + 5
-	Pengurangan	3 – 1
*	Perkalian	4 * 2
/	Pembagian	8 / 4
%	Sisa Pembagian (mod)	5 % 2

Sumber : www.duniailkom.com

Tabel II.5

Tabel Operator *Unary*

Operator	Keterangan	Contoh
+	Tanda Plus	+5
-	Tanda Minus	-5

Sumber : www.duniailkom.com

b. Operator Relasi / Perbandingan

Operator ini digunakan untuk membandingkan dua buah nilai dengan hasil berupa nilai numerik 1 (*True*) atau 0 (*False*).

Tabel II.6

Tabel Operator Relasi

Operator	Keterangan
==	Sama dengan (bukan pemberi nilai)
!=	Tidak Sama dengan
>	Lebih dari
<	Kurang dari
>=	Lebih dari atau Sama dengan
<=	Kurang dari atau Sama dengan

Sumber : www.duniailkom.com

c. Operator Logika / Boolean

Digunakan untuk mengekspresikan satu atau lebih data atau ekspresi logika (boolean) dengan menghasilkan data logika (boolean) baru.

Tabel II.7.
Operator Logika

Operator	Keterangan
&&	Operator Logika AND
	Operator Logika OR
!	Operator Logika NOT

Sumber : www.duniailkom.com

Operator Logika AND digunakan untuk menghubungkan dua atau lebih ekspresi relasi, akan bernilai benar jika semua ekspresi relasi yang dihubungkan bernilai benar. Operator Logika OR digunakan untuk menghubungkan dua atau lebih ekspresi relasi, akan bernilai benar jika salah satu ekspresi relasi yang dihubungkan bernilai benar dan akan bernilai salah jika semua ekspresi relasi yang dihubungkan bernilai salah. Operator Logika NOT akan memberikan nilai kebalikan dari ekspresi yang dihubungkan, jika ekspresi bernilai benar maka akan menghasilkan nilai salah, begitu dengan sebaliknya.

7. Konstanta

Adalah nilai yang telah ditentukan yang digunakan untuk membuat program lebih mudah dibaca. Konstanta dikelompokkan menjadi sebagai berikut:

a. *TRUE/FALSE*

Adalah konstanta boolean yang mendefinisikan level logika.

FALSE didefinisikan sebagai 0 (nol), sedangkan *TRUE* didefinisikan sebagai 1 (satu).

If (b == TRUE);

{ pernyataan; }

b. *HIGH/LOW*

Konstanta ini menentukan level tinggi atau rendah dan digunakan saat membaca atau menulis ke pin digital. *HIGH* didefinisikan sebagai logika 1, *on*, atau 5 *volt*. Sedangkan *LOW* didefinisikan sebagai logika 0, *off*, atau 0 *volt*.

digitalwrite(13, HIGH);

c. *INPUT/OUTPUT*

Konstanta ini digunakan dengan fungsi *pinMode()* untuk menentukan mode pin digital sebagai *input* atau *output*.

pinMode(13, OUTPUT);

8. Struktur Pengaturan

Sebuah program mempunyai suatu struktur dan pengaturan untuk menjalankan perintah programnya, yang biasa disebut juga dengan seleksi kondisi. Beberapa seleksi kondisi yang biasa digunakan dalam bahasa C sebagai berikut:

a. Perintah *If*

Digunakan untuk memilih satu dari dua atau lebih pernyataan.

Format: *if (kondisi)*

{ pernyataan }

b. Perintah *If...Else*

Digunakan untuk memilih satu dari dua atau lebih pernyataan dengan menjalankan pernyataan *if* terlebih dahulu, jika kondisinya *TRUE* maka akan diteruskan perintah programnya, jika kondisinya *FALSE* maka pernyataan *else* yang akan dijalankan.

Format: *if (kondisi) { Pernyataan }*

Else { pernyataan }

c. Perintah *If* dalam *If*

Perintah *if* dalam *if* sering disebut *nasted-if*, dengan menjalankan kondisi 1 *if* terlebih dahulu kemudian meneruskannya ke kondisi 2 *if* yang jika pernyataan 1 benar/*TRUE* maka diteruskan perintah programnya, jika kondisinya *FALSE* maka pernyataan 2 yang akan dijalankan.

Format: *if (kondisi1)*

{ if (kondisi2) { pernyataan1 }

Else { pernyataan2 } }

d. Perintah *For*

Digunakan untuk melakukan perulangan pada suatu kondisi menggunakan perhitungan (*counter*) yang pasti.

Format: *for (inisialisasi; syarat perulangan; pengubah nilai pencacah)*

{ pernyataan; }

Inisialisasi merupakan bagian untuk memberikan nilai awal untuk variabel-variabel tertentu. Syarat Perulangan yaitu memegang kontrol terhadap perulangan, karena bagian ini yang akan menentukan suatu

perulangan diteruskan atau dihentikan. Pengubah Nilai Pencacah; mengatur kenaikan atau penurunan nilai pencacah.

9. Digital I/O

I/O digital pada Arduino mempunyai fungsi sendiri untuk mengaksesnya. Fungsi ini menyederhanakan perintah yang berhubungan dengan pin I/O Arduino.

a. pinMode (pin, mode)

Berfungsi untuk mengkonfigurasi pin tertentu dan digunakan sebagai *input* atau *output*. Pin adalah nomor pin yang akan digunakan dari 0-19 (pin analog 0-5 adalah 14-19).

b. digitalWrite (pin, value)

Berfungsi untuk memberi nilai *HIGH* atau *LOW* ke pin digital. Jika pin dikonfigurasi sebagai *OUTPUT* dengan pinMode(), maka tegangan akan diatur ke nilai yang sesuai, yaitu 5V atau 3.3V untuk *HIGH* dan 0V (*ground*) untuk *LOW*. Jika pin dikonfigurasi sebagai *INPUT* dengan pinMode(), maka digitalWrite() akan mengaktifkan (*HIGH*) atau menonaktifkan (*LOW*) dengan pullup internal pada pin *input*.

c. digitalRead(pin)

Berfungsi untuk membaca nilai dari pin digital tertentu, baik itu *HIGH* atau *LOW*.

10. Analog I/O

Berikut ini adalah fungsi analog dari Arduino:

a. AnalogRead(pin)

Berfungsi untuk membaca nilai analog melalui pin analog. Pada Arduino Uno sendiri memiliki 6 *channel* analog. Ketika pin analog

ditetapkan sebagai *INPUT*, maka keluaran voltase dapat dibaca, yang keluarannya antara angka 0 (0V) dan 1024 (5V).

b. `AnalogWrite(pin, value)`

Pin pada Arduino ada yang mendukung *PWM (pulse width modulation)* yaitu pin 3, 5, 6, 9, 10, 11, yang dapat merubah pin hidup (*on*) atau mati (*off*) dengan cepat sehingga dapat membuatnya berfungsi seperti keluaran analog. *Value* atau nilai pada format kode tersebut adalah antara angka 0.

(0% duty cycle – 0V) dan 255 (100% duty cycle – 5V).

11. *Delay(time)*

Fungsi ini untuk menghentikan sementara program selama waktu yang ditentukan dalam milidetik. 1000 sama dengan 1 detik

```
Delay(1000);           //tunggu 1 detik
```

2.2.2. *Software Editor*

1. Arduino *IDE*

Menurut (ANANDA, 2018) mengatakan bahwa, “Arduino *IDE* (*Integrated Development Program*) merupakan *software* yang dipakai untuk membuat *sketch* yang akan di *upload* ke *board* arduino sendiri”.

Arduino *IDE* juga tergolong bentuk *software* pengembang program yang terintegrasi sehingga berbagai *tools* atau tampilan disediakan dan dinyatakan dalam bentuk antarmuka berbasis menu. Dengan menggunakan Arduino *IDE*, kesalahan penulisan *sketch* atau kebenaran penulisan *sketch* langsung dapat dibuktikan.



Sumber : <https://www.arduino.cc>

Gambar II.27

Arduino IDE

2. Bagian-bagian Arduino IDE

Berikut adalah fungsi *toolbar* yang terdapat pada Arduino IDE :



Verify : Berfungsi untuk melakukan *checking* kode yang dibuat pada kesalahan kompilasi.



Upload : Berfungsi untuk melakukan kompilasi program dan menguploadnya ke *Board* arduino yang sudah dikonfigurasi.



New : Berfungsi untuk membuat *sketch* baru.



Open : Berfungsi untuk membuka *sketch* yang sudah pernah disimpan.



Save : Berfungsi untuk menyimpan *sketch* yang telah dibuat.



Serial Monitor : Berfungsi untuk membuka *serial monitor*. *Serial*

monitor merupakan jendela yang menampilkan data apa saja yang dikirimkan atau dipertukarkan antara arduino dengan *sketch* pada port *serialnya*. *Serial monitor* ini dapat digunakan untuk menampilkan nilai proses, nilai pembacaan, dan pesan *error*. Perintah tambahan dapat ditemukan dalam lima menu: ***File, Edit, Sketch, Tool, Help***.

3. Menulis *Sketch*

Jalankan program arduino dari file master arduino yang kita simpan ataupun dari ikon di komputer desktop. Kemudian akan muncul program arduino *IDE* dengan tampilan *sketch* sesuai dengan tanggal dan bulan pada saat program dijalankan. *Sketch* atau kode pemrograman Arduino ditulis dibagian editor *text* dan disimpan dalam file dengan ekstensi *.ino*.

4. Verifikasi *Sketch*

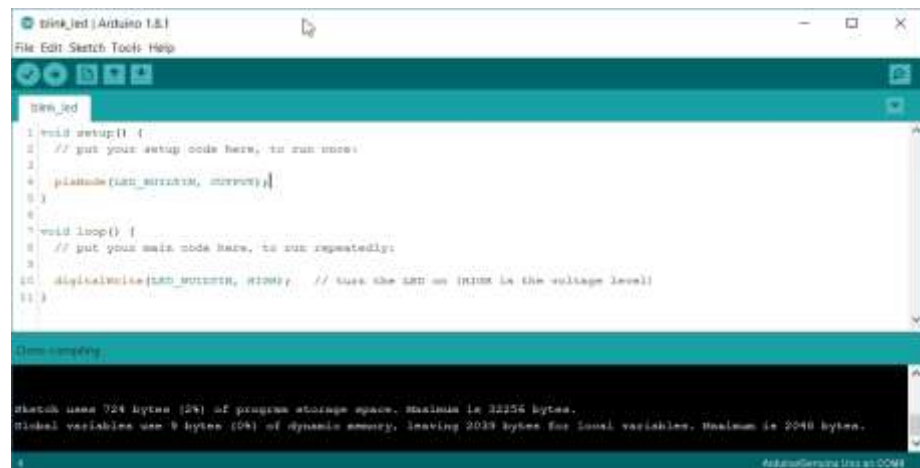
Jika *sketch* telah disimpan selanjutnya adalah verifikasi memastikan bahwa kode program telah benar. Klik tombol verifikasi untuk melakukannya, dan jendela pesan akan muncul “*Done Compiling*” jika verifikasi sukses.

a. Memilih Jenis dan Port *Board* Arduino

Sebelum mengupload *sketch* ke *board* arduino, diharuskan memilih jenis dan port *board* Arduino yang digunakan. Menu ***tools > Board*** untuk memilih *board* yang sesuai, menu ***tools > Port*** untuk memilih port yang digunakan. Pada Windows kemungkinan **COM1** dan **COM2** untuk port *serial* atau COM4, COM5, COM7, dan lebih tinggi untuk port *USB*. Untuk mengetahuinya, dapat dilihat perangkat *USB serial* pada seksi port dari *Windows Device Manager*.

b. Mengunggah *Sketch*

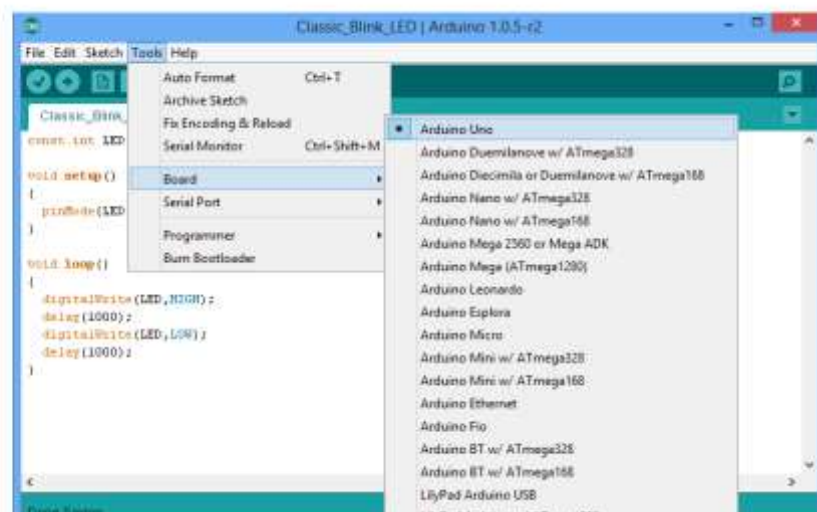
Klik tombol *Upload*, *LED* RX dan TX di *board* akan berkedip dan jendela *Arduino IDE* akan memberi status bar yang menunjukkan berapa banyak kemajuan yang telah diunggah. Jika unggahan berhasil, pesan “*Done uploading*”, akan muncul di jendela pesan.



Sumber : <http://www.circuitstoday.com/blink-led-with-arduino>

Gambar II.28

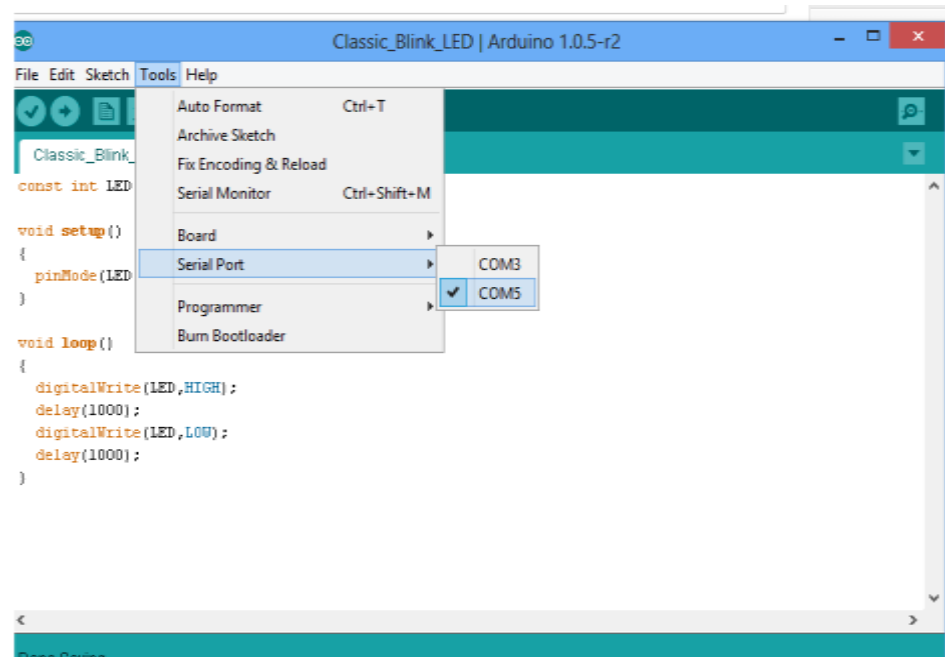
Verifikasi *Sketch*



Sumber : <http://www.circuitstoday.com/blink-led-with-arduino>

Gambar II.29

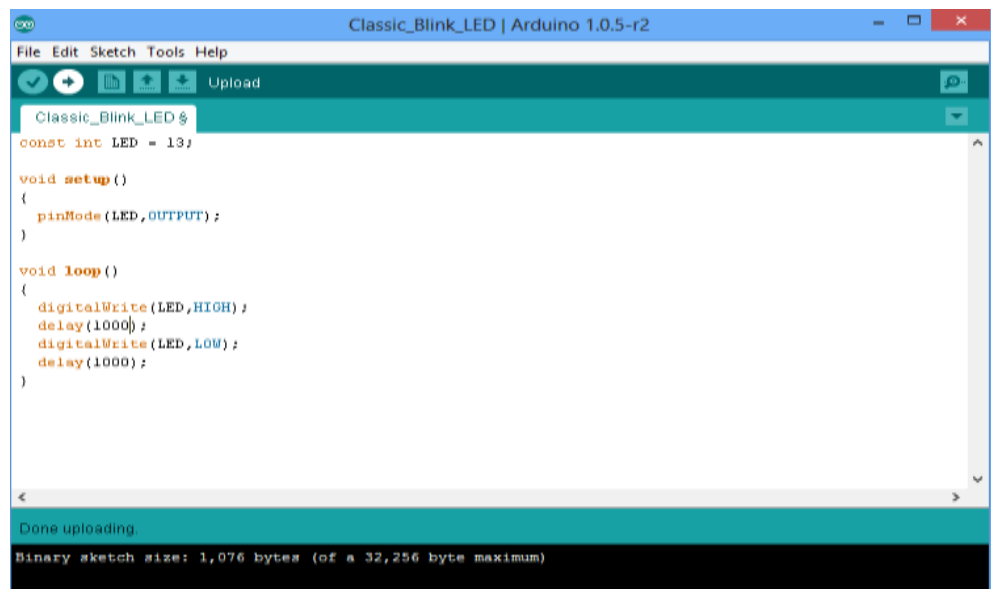
Memilih *Board*



Sumber : <http://www.circuitstoday.com/blink-led-with-arduino>

Gambar II.30

Memilih Port



Sumber : <http://www.circuitstoday.com/blink-led-with-arduino>

Gambar II.31

Sukses Mengunggah *Sketch*