

BAB II

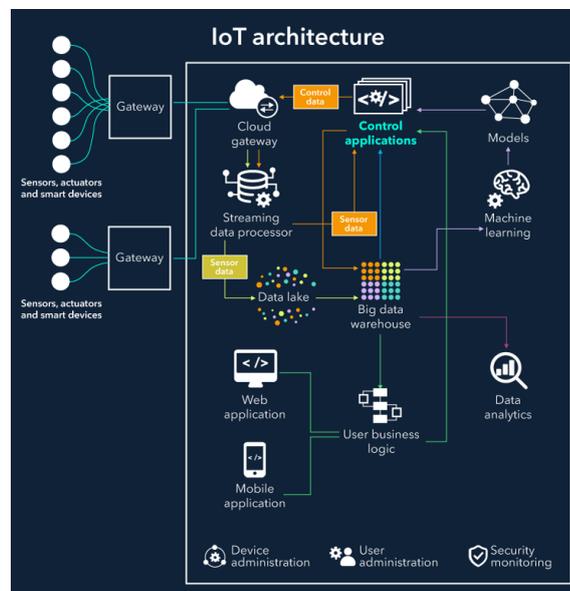
TINJAUAN PUSTAKA

1.1 Arsitektur *Internet of Things* (IoT)

Konsep IoT pertama kali dicetuskan oleh komunitas pengembang *Radio Frequency Identification* (RFID) sekitar tahun 1999. Konsep ini makin relevan dengan masa sekarang mengingat makin banyaknya pertumbuhan perangkat baik berupa telepon pintar, perangkat tertanam, sensor dan komputasi awan. Perangkat sensor dapat menangkap kondisi lingkungan dan berkomunikasi dengan menyebarkan informasi tersebut ke berbagai perangkat lain.

Kemudian, informasi tersebut bisa digunakan oleh sistem lain untuk menganalisa perilaku menentukan keputusan yang harus diambil. Dengan konsep dan skema tersebut dapat dibuat sebuah sistem yang cerdas.

Model komunikasi IoT dapat berupa komunikasi *devices to devices*, model ini menghubungkan dua atau lebih perangkat yang saling terhubung dan berkomunikasi langsung tanpa harus melewati server atau perangkat penghubung lain (Sukaridhoto & ST Ph, 2016).

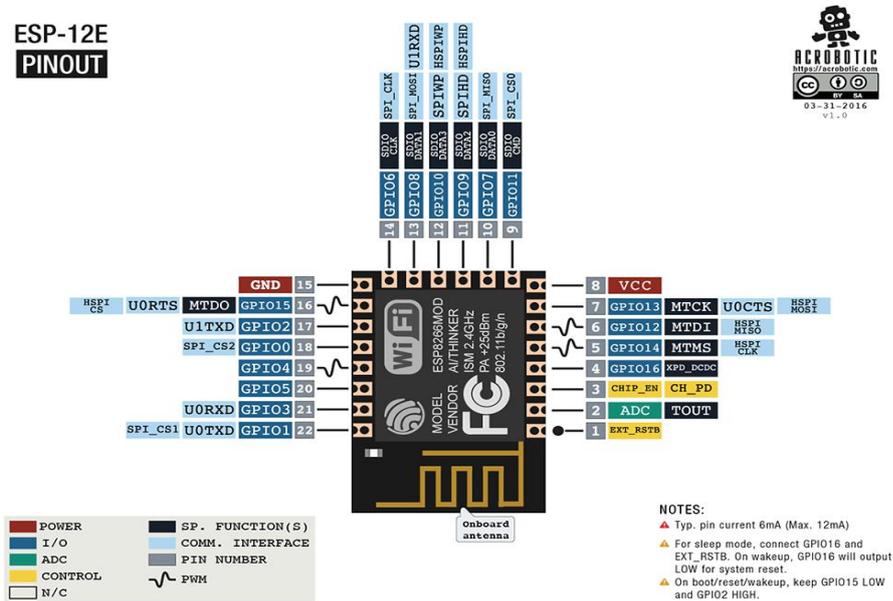


Gambar 2. 1 Arsitektur *Internet of Things* (IoT)
(Sumber : www.scnsoft.com)

1.2 NodeMCU ESP8266

Menurut (Wicaksono, 2017) *NodeMCU* adalah sebuah *platform* IoT yang bersifat *open-source* yang terdiri dari perangkat keras berupa *System On Chip* ESP8266 buatan *Espressif System*. *Firmware* yang digunakan adalah bahasa pemrograman *scripting* Lua. Istilah *nodeMCU* secara *default* sebenarnya mengacu pada *firmware* yang digunakan daripada perangkat keras *development kit*. *NodeMCU* bisa dianalogikan sebagai *board* arduinonya ESP8266.

Pada *nodeMCU* telah menanamkan *chip* wifi ESP8266 menjadi sebuah *board* yang memiliki fitur seperti *board* mikrokontroler *arduino*. *Board* ini juga dilengkapi *chip* CH340 untuk komunikasi serial atau sering disebut *USB to serial* sehingga hanya membutuhkan kabel USB *type B* sebagai kabel data. Kabel ini mirip kabel *charging smartphone android*.



Gambar 2. 2 Konfigurasi ESP8266

Espressif Systems pertama kali merilis modul *wifi* ESP8266 pada 30 Desember 2013. *NodeMCU* pertama kali dirilis pada 13 Oktober 2014 oleh *Espressif Systems*, perusahaan yang juga mengembangkan ESP8266. Dua bulan kemudian proyek tersebut dikembangkan ke platform perangkat keras ketika Huang R mengunduh *file* dari papan ESP8266, yang diberi nama devkit v.0.9.

Pada bulan oktober 2014 juga *library client* MQTT dari *Contiki* ke *platform* ESP8266 untuk *board nodeMCU* sehingga *nodeMCU* mendukung protokol IoT MQTT. Pada Januari 2015 *library LCD* untuk *nodeMCU* sehingga memudahkan *board nodeMCU* menampilkan data melalui LCD. Saat ini *library* untuk *nodeMCU* mencapai 40 *library* yang bisa digunakan sesuai kebutuhan pengembang.

ESP8266 adalah inti dari *nodeMCU* maka fitur – fitur *nodeMCU* kurang lebih sama seperti ESP8266. Pada bahasa pemrograman eLua yang mirip dengan *javascript nodeMCU* telah dibungkus oleh API. Perbedaan *nodeMCU* dengan ESP8266 adalah *nodeMCU* memiliki *chip* komunikasi serial sehingga dapat langsung berkomunikasi dengan PC/laptop saat melakukan pemrograman.

Pada gambar modul ESP diatas terdapat penjelasan pin tersebut, VCC yang berfungsi sebagai sumber dengan memberikan muatan positif, GND yang harus dihubungkan dengan muatan negatif, serta pin GPI yang berjumlah 13 buah pin yang berfungsi sebagai pin *input* maupun *output*, TX yang berupa pin *transmitter* dan RX sebagai pin *receiver* berfungsi dalam pengiriman dan penerimaan data program.

1.3 Mikrokontroler

Mikrokontroler adalah suatu chip berupa IC (*Integrated Circuit*) yang dapat menerima sinyal input, mengolahnya dan memberikan sinyal output sesuai dengan program yang diisikan ke dalamnya. Sinyal input mikrokontroler berasal dari sensor yang merupakan informasi dari lingkungan sedangkan sinyal output ditujukan kepada aktuator yang dapat memberikan efek ke lingkungan. Jadi secara sederhana mikrokontroler dapat diibaratkan sebagai otak dari suatu perangkat/produk yang mampu berinteraksi dengan lingkungan sekitarnya.

Mikrokontroler pada dasarnya adalah komputer dalam satu chip, yang di dalamnya terdapat mikroprosesor, memori, jalur Input/Output (I/O) dan perangkat pelengkap lainnya. Kecepatan pengolahan data pada mikrokontroler lebih rendah jika dibandingkan dengan PC (*Personal Computer*). Pada PC kecepatan mikroprosesor yang digunakan saat ini telah mencapai orde GHz, sedangkan kecepatan operasi mikrokontroler pada umumnya berkisar antara 1 – 16 MHz.

Begitu juga kapasitas RAM dan ROM pada PC yang bisa mencapai orde Gbyte, dibandingkan dengan mikrokontroler yang hanya berkisar pada orde byte/Kbyte.

Meskipun kecepatan pengolahan data dan kapasitas memori pada mikrokontroler jauh lebih kecil jika dibandingkan dengan komputer personal, namun kemampuan mikrokontroler sudah cukup untuk dapat digunakan pada banyak aplikasi terutama karena ukurannya yang kompak. Mikrokontroler sering digunakan pada sistem yang tidak terlalu kompleks dan tidak memerlukan kemampuan komputasi yang tinggi.

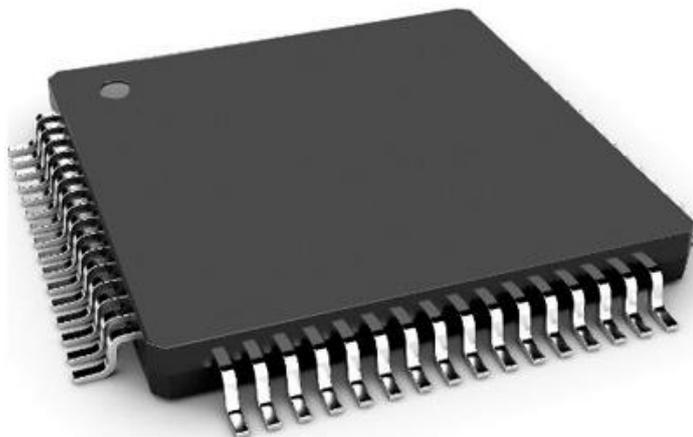
Sistem yang menggunakan mikrokontroler sering disebut sebagai *Embedded System* atau *Dedicated System* (Discription et al., 2013). *Embedded system* adalah sistem pengendali yang tertanam pada suatu produk, sedangkan *dedicated system* adalah sistem pengendali yang dimaksudkan hanya untuk suatu fungsi tertentu. Sebagai contoh, printer adalah suatu *embedded system* karena di dalamnya terdapat mikrokontroler sebagai pengendali dan juga *dedicated system* karena fungsi pengendali tersebut berfungsi hanya untuk menerima data dan mencetaknya. Hal ini berbeda dengan suatu PC yang dapat digunakan untuk berbagai macam keperluan, sehingga mikroprosesor pada PC sering disebut sebagai *General Purpose Microprocessor* (mikroprosesor serba guna). Pada PC berbagai macam software yang disimpan pada media penyimpanan dapat dijalankan, tidak seperti mikrokontroler hanya terdapat satu software aplikasi.

Penggunaan mikrokontroler antara lain terdapat pada bidang-bidang berikut ini:

1. Otomotif: *Engine Control Unit, Air Bag, Fuel Control, Antilock Braking System*, sistem pengaman alarm, transmisi otomatis, hiburan, pengkondisi udara, *speedometer* dan odometer, navigasi, suspensi aktif.
2. Perlengkapan Rumah Tangga dan Perkantoran: Sistem pengaman alarm, *remote control*, mesin cuci, *microwave*, pengkondisi udara, timbangan digital, mesin foto kopi, printer, *mouse*.
3. Pengendali Peralatan di Industri.
4. Robotika.

Saat ini mikrokontroler 8bit masih menjadi jenis mikrokontroler yang paling populer dan paling banyak digunakan. Maksud dari mikrokontroler 8bit adalah data yang dapat diproses dalam satu waktu adalah 8 bit, jika data yang diproses lebih besar dari 8 bit maka akan dibagi menjadi beberapa bagian data yang masing-masing terdiri dari 8 bit. Masing-masing mikrokontroler mempunyai cara dan bahasa pemrograman yang berbeda, sehingga program untuk suatu jenis mikrokontroler tidak dapat dijalankan pada jenis mikrokontroler lain. Untuk memilih jenis mikrokontroler yang cocok dengan aplikasi yang dibuat terdapat tiga kriteria yaitu:

1. Dapat memenuhi kebutuhan secara efektif & efisien. Hal ini menyangkut kecepatan, kemasan/packaging, konsumsi daya, jumlah RAM dan ROM, jumlah I/O dan timer, harga per unit.
2. Bahasa pemrograman yang tersedia.
3. Kemudahan dalam mendapatkannya.



Gambar 2. 3 Chip Mikrokontroler

1.4 Liquid Crystal Display (LCD)

(Aly Afandi et al., 2021a) LCD (*Liquid Cristal Display*) adalah salah satu jenis *display* elektronik yang dibuat dengan teknologi CMOS *logic* yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekelilingnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-lit*. LCD

berfungsi sebagai penampil data baik dalam bentuk karakter, huruf, angka ataupun grafik.

LCD adalah lapisan dari campuran organik antara lapisan kaca bening dengan elektroda transparan indium oksida dalam bentuk tampilan *seven-segment* dan lapisan elektroda pada kaca belakang. Ketika elektroda diaktifkan dengan medan listrik (tegangan), molekul organik yang panjang dan silindris menyesuaikan diri dengan elektroda dari segmen. Lapisan *sandwich* memiliki *polarizer* cahaya vertikal depan dan *polarizer* cahaya horizontal belakang yang diikuti dengan lapisan *reflektor*. Cahaya yang dipantulkan tidak dapat melewati molekul-molekul yang telah menyesuaikan diri dan segmen yang diaktifkan terlihat menjadi gelap dan membentuk karakter data yang ingin ditampilkan.

Menurut Muhammad Royhan (2018:34) Layar LCD merupakan suatu media penampilan data yang sangat efektif dan efisien dalam penggunaannya. Untuk menampilkan sebuah karakter pada layar LCD diperlukan beberapa rangkaian tambahan. Untuk lebih memudahkan para pengguna, maka beberapa perusahaan elektronik menciptakan modul LCD Tampilan LCD untuk menampilkan angka atau teks. dua jenis LCD Display. LCD yang digunakan untuk tampilan pengaturan menggunakan LCD 16x2 LCD dibagi menjadi dua bagian yaitu bagian depan panel LCD yang terdiri dari banyak dot atau titik LCD dan mikrokontroler yang menempel pada bagian belakang panel LCD yang berfungsi untuk mengatur titik-titik LCD sehingga dapat menampilkan huruf, angka, dan simbol khusus yang dapat terbaca.

2.4.1. Karakteristik LCD 16x2

1. 16 karakteristik x 2 baris
2. 5x7 titik matriks karakter + kursor
3. HD44780 *equivalent LCD controller/diver built-in*
4. 4 bit atau 8 bit MPU *interface* Tipe standar
5. Bekerja hampir semua mikrokontroler (Muhamad Royhan,2018:35)



Gambar 2. 4 Bentuk Fisik LCD 16 x 2

2.4.2. Spesifikasi LCD 16x2

Tabel 2. 1 Spesifikasi LCD 16x2

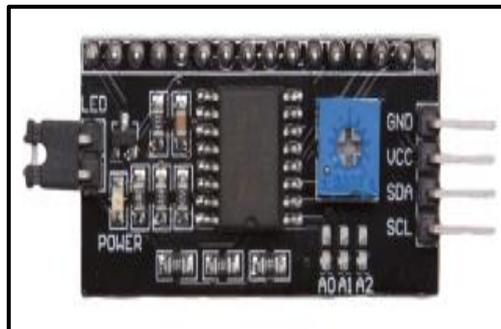
Pin	Simbol	Fungsi
1	Vss	Ground
2	Vdd	+3V Atau +5V
3	Vo	Pengatur Kontras
4	Rs	H/L Register Select Signal
5	R/W	Read/Write Signal
6	EN	Enable Signal
7-14	Data	I/O Pins
15	Anoda	Tegangan Positif
16	Katoda	Tegangan Negatif

1.5 *Inter Integrated Circuit (I2C)*

Menurut M.Natsir,dkk (2019:71) (Aly Afandi et al., 2021b) I2C/TWI LCD merupakan modul yang dipakai untuk mengurangi penggunaan kaki di LCD. Modul

ini memiliki 4 pin yang akan dihubungkan ke Arduino. Arduino uno sudah mendukung komunikasi I2C dengan module I2C lcd, maka dapat mengontrol LCD Karakter 16x2 dan 20x4 hanya menggunakan 2 Pin yaitu Analog *Input* Pin 4 (SDA) dan Analog *Input* Pin 5 (SCL).

Menurut Yohanes C Saghoa,dkk (2018:169) (Yohanes et al., 2018) *Inter Integrated Circuit* atau sering disebut I2C adalah standar komunikasi serial dua arah menggunakan dua saluran yang didesain khusus untuk mengirim maupun menerima data. 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*.

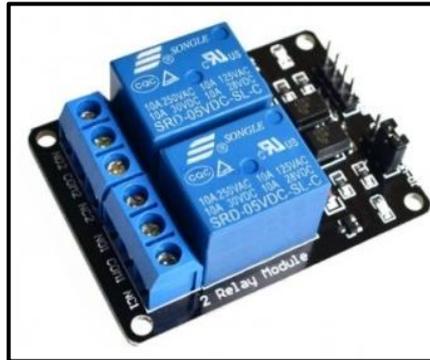


Gambar 2. 5 Inter Integrated Circuit (I2C)

1.6 Relay Modul

Relay adalah Saklar (*Switch*) yang dioperasikan secara listrik dan merupakan komponen *Electromechanical* (Elektromekanikal) yang terdiri dari 2 bagian utama yakni *Electromagnet (Coil)* dan Mekanikal (seperangkat Kontak Saklar/*Switch*).

Relay menggunakan Prinsip Elektromagnetik untuk menggerakkan Kontak Saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi.

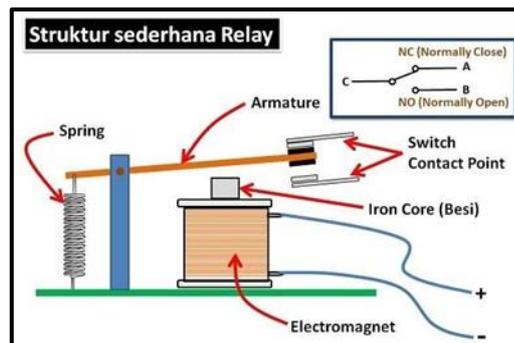


Gambar 2. 6 Relay

Pada dasarnya, Relay terdiri dari 4 komponen dasar yaitu :

1. *Electromagnet (Coil)*
2. *Armature*
3. *Switch Contact Point (Saklar)*
4. *Spring*

Berikut ini merupakan gambar dari bagian-bagian Relay :



Gambar 2. 7 Bagian-bagian Relay

Kontak Poin (*Contact Point*) Relay terdiri dari 2 jenis yaitu:

1. *Normally Close (NC)* yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi *CLOSE* (tertutup).
2. *Normally Open (NO)* yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi *OPEN* (terbuka).

Karena Relay merupakan salah satu jenis dari Saklar, maka istilah *Pole* dan *Throw* yang dipakai dalam Saklar juga berlaku pada Relay. Berikut ini adalah penjelasan singkat mengenai Istilah *Pole* and *Throw* :

1. *Pole* : Banyaknya Kontak (*Contact*) yang dimiliki oleh sebuah relay
2. *Throw* : Banyaknya kondisi yang dimiliki oleh sebuah Kontak (*Contact*)

Berdasarkan penggolongan jumlah *Pole* dan *Throw*-nya sebuah relay, maka relay dapat digolongkan menjadi :

1. *Single Pole Single Throw (SPST)* : Relay golongan ini memiliki 4 Terminal, 2 Terminal untuk Saklar dan 2 Terminalnya lagi untuk *Coil*.
2. *Single Pole Double Throw (SPDT)* : Relay golongan ini memiliki 5 Terminal, 3 Terminal untuk Saklar dan 2 Terminalnya lagi untuk *Coil*.
3. *Double Pole Single Throw (DPST)* : Relay golongan ini memiliki 6 Terminal, diantaranya 4 Terminal yang terdiri dari 2 Pasang Terminal Saklar sedangkan 2 Terminal lainnya untuk *Coil*. Relay DPST dapat dijadikan 2 Saklar yang dikendalikan oleh 1 *Coil*.
4. *Double Pole Double Throw (DPDT)* : Relay golongan ini memiliki Terminal sebanyak 8 Terminal, diantaranya 6 Terminal yang merupakan 2 pasang Relay SPDT yang dikendalikan oleh 1 (*single*) *Coil*. Sedangkan 2 Terminal lainnya untuk *Coil*.

Seperti yang telah dijelaskan tadi bahwa relay memiliki fungsi sebagai saklar elektrik, namun jika di aplikasikan ke dalam rangkaian elektronika, relay memiliki beberapa fungsi yang cukup unik. Berikut beberapa fungsi saat di aplikasikan ke dalam sebuah rangkaian elektronika.

1. Relay digunakan untuk menjalankan Fungsi Logika (*Logic Function*)
2. Relay digunakan untuk memberikan Fungsi penundaan waktu (*Time Delay Function*)
3. Relay digunakan untuk mengendalikan Sirkuit Tegangan tinggi dengan bantuan dari Signal Tegangan rendah (Muhamad Saleh, dan Munnik Haryanti, 2017:181). (Sirait & Wicaksono, 2017)

1.7 Arduino IDE

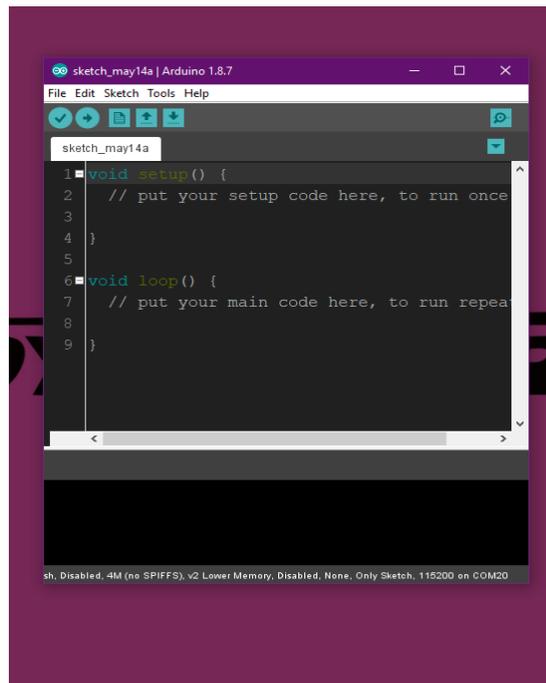
Banyak bahasa pemrograman yang biasa digunakan untuk pemrograman mikrokontroler contohnya adalah bahasa basic. Akan Tetapi pemrograman yang dipakai pada *Arduino IDE* adalah bahasa C. Bahasa C adalah bahasa yang *simple* sehingga mudah diaplikasikan dan mudah untuk dipelajari. Bahasa C sangat berperan pada perkembangan teknologi sampai saat ini khususnya perkembangan *software – software* komputer.

Operation system dan *compiler* yang populer saat ini rata – rata menggunakan bahasa C untuk bahasa pemrogramannya. Contohnya adalah *codevisionAVR*, keil *compiler*, dan *visual studio*. Bahasa C adalah bahasa pemrograman yang sangat ampuh kekuatannya mendekati bahasa *assembler*. Bahasa C yang di *compile* menghasilkan *file* kode objek yang sangat kecil dan dieksekusi dengan sangat cepat. Bahasa C sering digunakan pada operasi dan pemrograman mikrokontroler.

Bahasa C adalah *multi-platform* karena Bahasa C bisa diterapkan pada lingkungan *Windows, Unix, Linux* atau sistem operasi lain tanpa mengalami perubahan *source code*. Jika ada perubahan, perubahannya sangat sedikit. Oleh karena itu *arduino* dapat dijalankan pada semua sistem operasi yang umum seperti *Windows, Linux* dan *Mac* (Syahwil, 2013).

Gambar 2.7 adalah tampilan dari *Arduino IDE* terdapat *void setup* dan *void loop*. *Arduino* menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Berikut kegunaan menu dari *Arduino IDE*

- A. *Verify* berfungsi mengecek kode atau *program* yang kita kerjakan sudah sesuai dengan bahasa pemrograman yang sudah ada atau belum.
- B. *Upload* berfungsi mengirim kode atau program yang telah kita kerjakan dapat dipahami oleh mikrokontroler *Arduino IDE* itu sendiri.
- C. *Serial monitor* berfungsi sebagai jendela yang menampilkan berupa data yang dikirim antara *arduino* dengan menu *sketch* pada *port* serial.
- D. *New* berfungsi membuat dan menampilkan *sketch* yang baru.



Gambar 2. 8 Tampilan Arduino IDE

Di internet banyak *library* Bahasa C untuk *Arduino* yang bisa di *download* secara gratis. Setiap *library Arduino* biasanya disediakan contoh atau *example*. Adanya *library – library* sangat membantu dalam proyek – proyek mikrokontroler. Selain itu, dapat dijadikan sarana untuk mendalami pemrograman Bahasa C pada *mikrokontroler*.

2.7.1. *Function Syntax Arduino*

Setiap program *Arduino* atau yang sering disebut *Sketch* mempunyai dua buah fungsi sintaks yang harus ada. Dua buah fungsi tersebut adalah:

a. *Void Loop ()*

Fungsi ini dijalankan setelah fungsi *setup* selesai dijalankan. Setelah selesai dijalankan satu kali fungsi *loop* ini akan kembali menjalankan fungsi *loop* secara terus – menerus sampai catu daya atau *power* dilepaskan.

b. *Void Setup ()*

Fungsi ini dijalankan pertama kali ketika *Arduino* dinyalakan. Fungsi ini hanya berjalan sekali. Berbeda dengan fungsi *loop* yang berjalan berkali – kali.

2.7.2. Operator Matematika

Operator yang digunakan untuk memanipulasi angka (bekerja seperti matematika sederhana). Berikut ini adalah operator matematika sederhana :

a. = (sama dengan)

Digunakan untuk membuat sesuatu *variable* bernilai sama dengan *variable* yang lainnya contohnya adalah $e = 12 * 2$, e sekarang sama dengan 24.

b. % (modulo)

Digunakan untuk mendapatkan sisa dari hasil pembagian. Contohnya adalah $25 \% 2$, ini akan menghasilkan angka 1.

c. + (penjumlahan)

Digunakan untuk menambahkan dua buah *variable* atau lebih. Contohnya adalah $x = 12 + 13 + 24$.

d. - (pengurangan)

Digunakan untuk mengurangi dua buah *variable* atau lebih. Contohnya adalah $x = 12 - 3 - 4$.

e. * (perkalian)

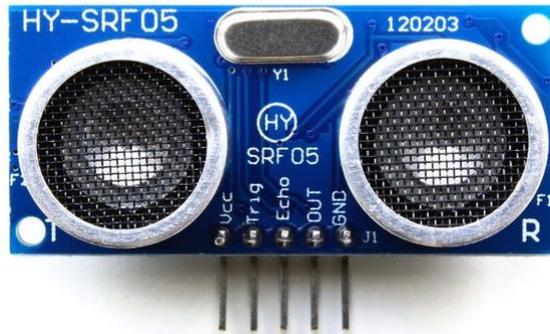
Digunakan untuk mengalikan dua buah *variable* atau lebih. Contohnya adalah $x = 12 * 3$.

f. / (pembagian)

Digunakan untuk membagi dua buah *variable* atau lebih. Contohnya adalah $x = 12 / 3$.

1.8 HY-SRF 05

HY-SRF 05 merupakan sensor ultrasonik siap pakai, satu alat yang berfungsi sebagai pengirim, penerima, dan pengontrol gelombang ultrasonik. Alat ini bisa digunakan untuk mengukur jarak benda dari 2cm - 4m dengan akurasi 3mm. Berikut Gambar dan spesifikasi sensor yang ditunjukkan pada Gambar 2.9 dan Tabel 2.2.



Gambar 2. 9 HY-SRF05

Tabel 2. 2 Spesifikasi sensor HY-SRF 05

No	Nama	Spesifikasi
1	Power Supply	+5V DC
2	Arus daya	15mA
3	Sudut efektif	<15o
4	Pembacaan jarak	2cm – 400cm
5	Pengukuran sudut	30o

1.9 Blynk

Blynk adalah sebuah layanan server yang digunakan untuk mendukung *project Internet of Things*. Layanan server ini memiliki lingkungan *mobile user* baik Android maupun iOS. Blynk Aplikasi sebagai pendukung IoT dapat diunduh melalui Google Play dan dapat mendukung berbagai *platform* kerja seperti pada mikrokontroler atau *Smarthome*.

Blynk adalah *dashboard* digital dengan fasilitas antarmuka grafis dalam pembuatan sistemnya, penambahan komponen pada Blynk Apps dilakukan dengan cara *Drag and Drop* sehingga memudahkan para penggunanya dalam menyesuaikan kebutuhan dan kegunaan dari sistem yang akan dibuat. Blynk diciptakan dengan tujuan untuk melakukan kontrol dan monitoring perangkat keras secara jarak jauh menggunakan berbagai macam media komunikasi mulai dari

Bluetooth, Wi-Fi, Ethernet, Jaringan LAN (*Local Acces Network*) sampai Koneksi data Internet Nirkabel (Yuan et al., 2012).

Kemampuan untuk menyimpan data dan menampilkan data secara visual baik menggunakan angka, warna ataupun grafik membuat aplikasi ini semakin banyak dipilih untuk mendukung sistem *Internet of Things*. Setidaknya, ada 3 komponen utama dalam penyusunan sistem aplikasi Blynk, meliputi:

1. Blynk Apps

Blynk Apps memungkinkan pengguna untuk membuat *project* dengan antarmuka yang tidak rumit atau *User-Friendly*, dengan berbagai macam komponen *input* serta *output* yang mendukung untuk pengiriman maupun penerimaan data serta merepresentasikan data sesuai dengan sumber data dari komponen yang dipilih. Representasi data dapat berbentuk visual angka, huruf, notifikasi maupun grafik. Terdapat 4 jenis kategori komponen yang berdatang pada Aplikasi Blynk

- A. *Controller* digunakan untuk mengirimkan data atau perintah ke *Hardware*
- B. *Display* digunakan untuk menampilkan data yang berasal dari *hardware* ke *Smartphone*
- C. *Notification* digunakan untuk mengirim pesan dan notifikasi.
- D. *Interface* Pengaturan tampilan pada aplikasi Blynk dapat berupa menu ataupun tab

2. Blynk Server

Blynk server merupakan fasilitas *Backend Service* berbasis *Cloud Server* (Komputasi Awan) yang bertanggung jawab untuk mengatur komunikasi antara aplikasi *smartphone* dengan lingkungan *hardware*.

Kemampuannya untuk terkoneksi dengan banyak perangkat atau *Multiple Acces Devices* pada saat yang bersamaan semakin memudahkan bagi para pengembang sistem *Internet of Things*. Blynk server juga tersedia dalam bentuk lokal server apabila digunakan pada lingkungan tanpa koneksi internet. Blynk

server lokal bersifat *open source* dan dapat diimplementasikan pada Hardware Raspberry Pi atau Mini Komputer.

3. Blynk Library

Blynk Library dapat digunakan untuk membantu pengembangan code. Blynk library tersedia pada banyak platform perangkat keras sehingga semakin memudahkan para pengembang IoT dengan fleksibilitas hardware yang didukung oleh lingkungan Blynk.



Gambar 2. 10 Blynk App

1.10 Flowchart

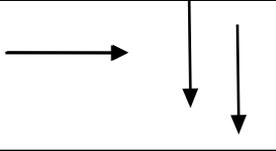
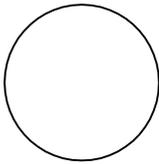
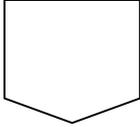
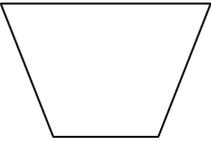
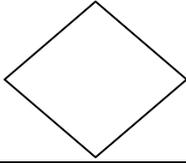
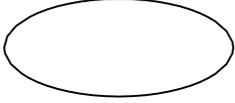
Menurut Wibawanto (2017) *flowchart* adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program. Diagram alur dapat menunjukkan secara jelas, arus pengendalian suatu algoritma yakni bagaimana melaksanakan suatu rangkaian kegiatan secara logis dan sistematis.

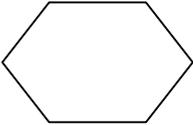
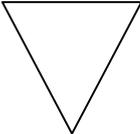
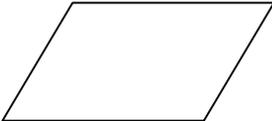
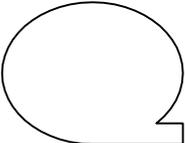
Simbol-simbol yang di pakai dalam flowchart dibagi menjadi 3 kelompok:

- 1) *Flow direction symbol*, digunakan untuk menghubungkan simbol satu dengan yang lain, disebut juga connecting line.
- 2) *Processing symbols*, Menunjukkan jenis operasi pengolahan dalam suatu proses/prosedur.
- 3) *Input/Output symbol*, menampilkan jenis peralatan yang digunakan sebagai media input atau output.

Berikut dibawah ini adalah simbol-simbol yang digunakan dalam flowchart disertai dengan keterangan fungsinya sebagaimana dijelaskan pada tabel 2.3.

Tabel 2. 3 Tabel *Flow Symbol*

NO	SIMBOL	KETERANGAN
1.		Simbol arus/ <i>flow</i> , berfungsi untuk menyatakan jalannya arus suatu proses
2.		Simbol <i>connector</i> , berfungsi untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
3.		Simbol <i>offline connector</i> , berfungsi untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda
4.		Simbol <i>process</i> , berfungsi untuk menyatakan suatu tindakan (proses) yang dilakukan oleh computer
5.		Simbol <i>manual</i> , berfungsi untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh computer
6.		Simbol <i>decision</i> , berfungsi untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya/tidak
7.		Simbol <i>teminal</i> , berfungsi untuk menyatakan permulaan atau akhir suatu program

8.		Simbol <i>predefined process</i> , berfungsi untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
9.		Simbol <i>keying operation</i> , berfungsi untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai <i>keyboard</i>
10.		Simbol <i>offline-storage</i> , berfungsi untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu
11.		Simbol <i>manual input</i> , berfungsi untuk memasukkan data secara manual dengan menggunakan <i>online keyboard</i>
12.		Simbol <i>input/output</i> , berfungsi untuk menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya
13.		Simbol <i>magnetic tape</i> , berfungsi untuk menyatakan <i>input</i> berasal dari pita magnetis atau <i>output</i> disimpan ke pita magnetis
14.		Simbol <i>disk storage</i> , berfungsi untuk menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i>
15.		Simbol <i>document</i> , berfungsi untuk mencetak keluaran dalam bentuk dokumen (melalui <i>printer</i>)

16.		Simbol <i>punched card</i> , berfungsi untuk menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu
-----	---	---

