

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Tabel Perbandingan yang Sejenis

Sebagai perbandingan untuk referensi penulis dalam membuat tugas akhir ini, penulis membandingkan dua penelitian yang sejenis sebagai berikut:

**Tabel 2.1** Perbandingan Penelitian yang Sejenis [1]

No.	Judul Jurnal	Nama Peneliti / Tahun	Teknologi yang dipakai	Keunggulan	Kelemahan
1.	Perancangan Prototype Alat Bajak Sawah Dengan Pengontrolan Menggunakan Bluetooth Berbasis Android	Arief Cahya Purnomo; 2020	Arduino Nano, Modul Bluetooth HC-05	Respon dari alat cepat karena menggunakan koneksi bluetooth yang tidak terganggu oleh kuat sinyal	Tidak ada penjelasan tentang jarak jangkauan yang dapat dicapai oleh modul bluetooth yang dipakai
2.	Perancangan dan Implementasi Golden Ratio Pada Desain Mekanik Dan Elektrik Aero Robotik	Suharmin Bahri, Mohamad Ramdhani, Unang Sunarya; 2016	<i>Electronic Speed Controler</i> (ESC), Golden Ratio, Tx/Rx Remote Control	Respon dari pergerakan alat simulasi tergolong cepat karena menggunakan respon dari TX/RX Remote Control	Getaran yang besar yang dihasilkan dari brushless yang tidak stabil dapat berdampak buruk terhadap pembacaan sensor robot

**Tabel 2.1** Perbandingan Penelitian yang Sejenis [1]

No.	Judul Jurnal	Nama Peneliti / Tahun	Teknologi yang dipakai	Keunggulan	Kelemahan
3.	Simulasi Robot Pembajak Sawah Berbasis IoT	Ahmad Ade Farisi, Arli Daffa Fernanda; 2020	Modul Wemos D1 ESP8266, Smartphone Android, Modul Motor Driver L298N	Dapat mengontrol alat dari jarak yang jauh, terdapat kontrol cadangan berupa remote, kontrol penggaruk / alat bajak dapat dilakukan melalui smartphone	Respon alat sangat bergantung pada kondisi sinyal karena berbasis IoT (menggunakan WiFi), biasanya terdapat delay respon pada alat setelah diberikan perintah gerak.

## 2.2 Pengenalan *Software*

### 2.2.1 Pengertian *Software*

Pada tahun 1970, kurang dari 1 persen masyarakat dapat dengan cerdas menggambarkan apa yang dimaksud dengan "perangkat lunak komputer". Saat ini, sebagian besar profesional dan banyak anggota masyarakat pada umumnya merasa bahwa mereka memahami perangkat lunak. Deskripsi buku teks tentang perangkat lunak dapat berbentuk sebagai berikut: "*Perangkat lunak adalah (1) instruksi (program komputer) yang ketika dijalankan memberikan fungsi dan kinerja yang diinginkan, (2) struktur data yang memungkinkan program untuk*

*memanipulasi informasi secara memadai, dan (3) dokumen yang menggambarkan operasi dan penggunaan program”*. Tidak ada pertanyaan bahwa definisi lain yang lebih lengkap dapat ditawarkan. Tetapi membutuhkan lebih dari sekadar definisi formal (Roger S. Pressman, Ph.D., Software Engineering, 2002: 6)

### **2.2.2 Karakteristik Software**

Untuk mendapatkan pemahaman tentang perangkat lunak (dan akhirnya pemahaman tentang rekayasa perangkat lunak), penting untuk memeriksa karakteristik perangkat lunak yang membuatnya berbeda dari hal-hal lain yang dibangun manusia. Ketika perangkat keras dibangun, proses kreatif manusia (analisis, desain, konstruksi, pengujian) pada akhirnya diterjemahkan ke dalam bentuk fisik. Jika kita membangun komputer baru, sketsa awal kita, gambar desain formal, dan prototipe papan tempat memotong roti berevolusi menjadi produk fisik (chip, papan sirkuit, catu daya, dll.).

Perangkat lunak lebih logis daripada elemen sistem fisik. Oleh karena itu, perangkat lunak memiliki karakteristik yang jauh berbeda dari perangkat keras:

1. Perangkat lunak dikembangkan atau direkayasa, tidak diproduksi dalam arti klasik.

Meskipun ada beberapa kesamaan antara pengembangan perangkat lunak dan pembuatan perangkat keras, kedua kegiatan ini pada dasarnya berbeda. Dalam kedua kegiatan, kualitas tinggi dicapai melalui desain yang baik, tetapi fase pembuatan untuk perangkat keras dapat menimbulkan masalah kualitas yang tidak ada (atau mudah diperbaiki) untuk perangkat lunak. Kedua kegiatan ini tergantung pada orang-orang, tetapi hubungan antara orang-orang yang diterapkan dan pekerjaan yang dicapai sama sekali berbeda. Kedua kegiatan tersebut membutuhkan pembangunan "produk" tetapi pendekatannya berbeda. Biaya perangkat lunak terkonsentrasi di bidang teknik. Ini berarti bahwa proyek perangkat lunak tidak dapat dikelola seolah-olah itu adalah proyek manufaktur.

## 2. Perangkat lunak tidak "aus."

Hubungan, sering disebut "kurva bak mandi," menunjukkan bahwa perangkat keras menunjukkan tingkat kegagalan yang relatif tinggi di awal kehidupannya (kegagalan ini sering dikaitkan dengan cacat desain atau manufaktur), cacat diperbaiki dan tingkat kegagalan turun ke tingkat kondisi mapan (idealnya, cukup rendah) untuk beberapa periode waktu. Namun seiring berjalannya waktu, tingkat kegagalan meningkat lagi karena komponen perangkat keras menderita dampak kumulatif debu, getaran, penyalahgunaan, suhu ekstrem, dan banyak penyakit lingkungan lainnya. Secara sederhana, perangkat keras mulai aus. Perangkat lunak tidak rentan terhadap penyakit lingkungan yang menyebabkan perangkat keras aus.

Maka dapat disimpulkan bahwa *software* merupakan rangkaian instruksi program yang diperlukan untuk menghasilkan informasi yang ditulis dengan bahasa khusus yang dimengerti oleh komputer dan manusia.

Dalam hal ini *software* dan bahasa pemrograman yang digunakan dalam pembuatan alat ini yaitu menggunakan Bahasa C, dan aplikasi yang digunakan yaitu Arduino IDE

### 2.3 *Internet of Things (IoT)*

#### 2.3.1 **Pengertian IoT**

IoT (*Internet of Things*) saat ini memang sedang marak dipakai di dunia IT. Pengertian dari IoT yaitu Sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus. Adapun kemampuan seperti berbagi data, *remote control*, dan sebagainya, termasuk juga pada benda di dunia nyata. Contohnya bahan pangan, elektronik yang terhubung ke jaringan lokal dan global melalui sensor yang terpasang dan selalu aktif (Wikipedia)

### 2.3.2 Contoh Aplikasi IoT

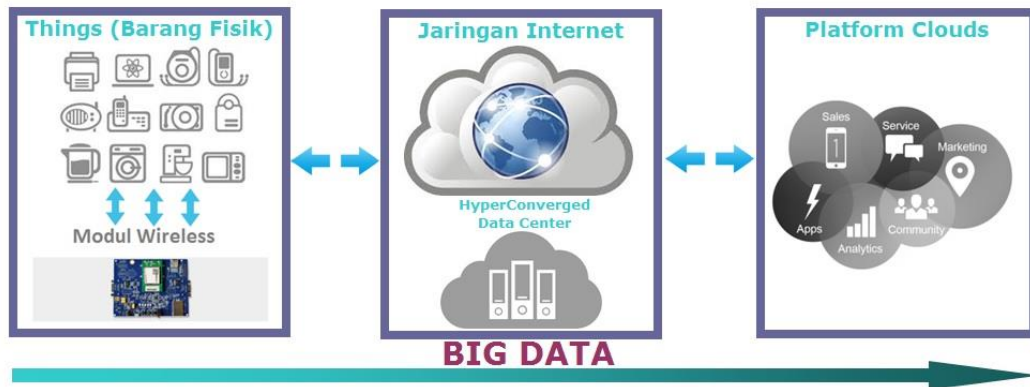
Banyak sekali contoh penerapan teknologi IoT, diantaranya yaitu:

1. Smart Home, sistem keamanan rumah berbasis internet, dapat mengetahui keadaan rumah serta mengontrol peralatan rumah tangga melalui jaringan internet.
2. Internet Industri, monitoring dan kontroling peralatan serta proses industri .
3. Pertanian, monitoring dan pengendalian peralatan pertanian.
4. Kesehatan, monitoring kondisi kesehatan seseorang.
5. Transportasi, manajemen dan informasi lalu lintas.

### 2.3.3 Sistem pada IoT

Sistem dasar dari IoT terdiri dari 3 hal:

1. Hardware / fisik (*Things*).
2. Koneksi internet.
3. *Cloud data centre* sebagai tempat tempat untuk menyimpan dan menjalankan aplikasi.



Gambar 2.1 Skema IoT [4]

## 2.4 WiFi

Sebuah firma konsultasi merk bernama *Interbrand Corporation* mencetuskan istilah WiFi, pada tahun 1999 tepatnya bulan Agustus istilah ini mulai dipakai. WiFi (*Wireless Fidelity*) adalah koneksi tanpa kabel seperti handphone dengan mempergunakan teknologi radio sehingga pemakainya dapat mentransfer data dengan cepat dan aman. WiFi tidak hanya dapat digunakan untuk mengakses internet, WiFi juga dapat digunakan untuk membuat jaringan tanpa kabel di perusahaan (WAN) (Hikmah Fajar Assidiq, 2019:15)

Berikut ini beberapa kegunaan WiFi selain digunakan untuk internet, yaitu:

1. Mensinkronisasi Smartphone dengan PC tanpa USB

WiFi bisa digunakan untuk sinkronisasi ke Smartphone dengan laptop/komputer. Sehingga memudahkan pemakai untuk tidak repot menggunakan kabel data maupun kabel USB. Tetapi masih harus menggunakan aplikasi-aplikasi tambahan dan masih terpaku pada beberapa jenis hp saja. Contohnya Smartphone Android dan Iphone, khusus Android harus ada aplikasi pihak ketiga seperti DoubleTwist.

2. Mengubah Smartphone menjadi *remote control*

Aplikasi Pihak ketiga seperti Telegram atau Blynk dapat di install pada smartphone yang berfungsi sebagai remote untuk mengoperasikan alat yang telah terintegrasi ke modul WiFi yang akan dikendalikan melalui smartphone. Smartphone berubah menjadi remote control untuk kendali yang diberikan oleh alat yang diperintahkan dengan intruksi tertentu.

3. Mentransfer foto dari kamera digital

Kartu Eye-Fi adalah kartu memori tanpa nirkabel. pada dasarnya seperti kartu SDHC, tetapi dengan manfaat yang besar ketika kamera sedang dalam jangkauan tertentu, foto dan video akan tertransfer ke komputer/laptop.

Berikut spesifikasi dari WiFi berdasarkan IEEE 802.11, ada empat variasi yaitu:

1. 802.11a

IEEE 802.11a adalah sebuah teknologi jaringan nirkabel yang merupakan pengembangan lebih lanjut dari standar IEEE 802.11 yang asli, namun bekerja pada bandwidth 5.8 GHz dengan kecepatan maksimum hingga 54 Mb/s. Metode transmisi yang digunakan adalah *Orthogonal Frequency Division Multiplexing* (OFDM), yang mengizinkan pentransmisi data secara paralel di dalam sub-frekuensi. Penggunaan OFDM memiliki keunggulan resistansi terhadap interferensi dengan gelombang lain, dan tentunya peningkatan throughput standar ini selesai diratifikasi pada tahun 1999.

2. 802.11b

IEEE 802.11b merupakan pengembangan dari standar IEEE 802.11 yang asli, yang bertujuan untuk meningkatkan kecepatan hingga 5.5 Mbps atau 11 Mbps tapi tetap menggunakan frekuensi 2.45 GHz. Dikenal juga dengan IEEE 802.11 HR. Pada praktek, kecepatan maksimum yang dapat diraih oleh standar IEEE 802.11b mencapai 5.9 Mbps pada protokol TCP, dan 7.1 Mbps pada protokol UDP. Metode transmisi yang digunakan adalah DSSS.

3. 802.11g

IEEE 802.11g adalah sebuah standar jaringan nirkabel yang bekerja pada frekuensi 2.45 GHz dan menggunakan metode modulasi OFDM. 802.11g yang dipublikasikan pada bulan juni 2003 mampu mencapai kecepatan hingga 54 Mbps pada pita frekuensi 2.45 GHz, sama sepenuhnya IEEE 802.11 biasa dan IEEE 802.11b. Standar ini menggunakan modulasi sinyal OFDM, sehingga lebih resistan terhadap interferensi dari gelombang lainnya. Kelebihan 802.11g yaitu kecepatan maksimum lebih cepat, jangkauan sinyal yang baik dan tidak mudah terhalang. Kekurangan 802.11g yaitu biaya yang lebih mahal dari 802.11b, peralatan dapat terganggu pada sinyal frekuensi yang tidak teratur.

#### 4. 802.11n

Standar jaringan wireless masa depan yang bekerja pada frekuensi 2.4 GHz dan dikabarkan berkecepatan transfer data mencapai 100-200 Mbps. Standar IEEE terbaru dalam kategori WiFi adalah 802.11n, variasi ini dirancang untuk memperbaiki fitur 802.11g dalam jumlah bandwidth yang didukung dengan memanfaatkan beberapa sinyal nirkabel dan antenna (disebut MIMO teknologi). Koneksi 802.11n harus mendukung kecepatan data yang lebih dari 100 Mbps. 802.11n juga menawarkan jangkauan yang lebih baik dari standar WiFi sebelumnya karena intensitas sinyal meningkat. Peralatan 802.11n akan kompatibel dengan alat-alat 802.11g. Keunggulan dari 802.11n yaitu kecepatan maksimum serta jangkauan sinyal tercepat dan terbaik, lebih tahan terhadap sinyal interferensi dari sumber-sumber luar. Kelemahan 802.11n yaitu, biaya lebih mahal dari 802.11g penggunaan beberapa sinyal akan sangat mungkin terganggu bila berdekatan dengan 802.11b/g berbasis jaringan.

**Tabel 2.2** Spesifikasi WiFi [2]

Spesifikasi	Kecepatan	Frekuensi Band	Kompatibel
802.11b	11 Mbps	~2.4 GHz	b
802.11a	54 Mbps	~2.4 GHz	a
802.11g	54 Mbps	~2.4 GHz	b,g
802.11n	100 Mbps	~2.4 GHz	b,g,n

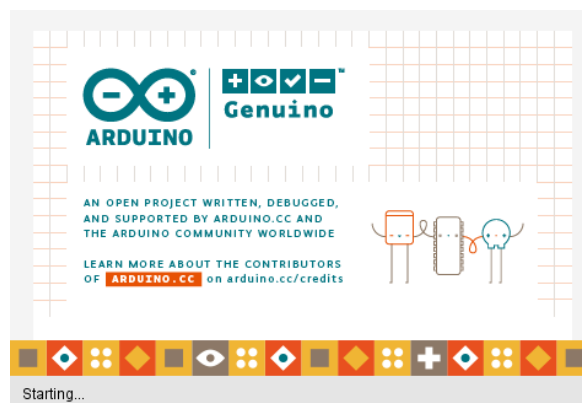
Untuk melakukan proses berjalannya sistem kontrol jarak jauh pada Robot Pembajak Sawah berbasis *Internet of Things* (IoT) maka diperlukan Modul WiFi untuk membantu transmisi data secara *Wireless*. Pada simulasi ini digunakan modul WiFi keluaran Arduino yaitu Wemos D1 ESP 8266.

#### 2.5 Arduino IDE

Arduino IDE adalah sebuah perangkat lunak pemrograman yang berfungsi untuk memasukkan program ke dalam sirkuit board modul Arduino salah satunya



Modul Wemos D1 ESP8266. IDE merupakan kependekan dari *Integrated Development Environment*, atau merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui software inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler. Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan library C/C++ yang biasa disebut Wiring yang membuat operasi input dan output menjadi lebih mudah. Arduino IDE ini dikembangkan dari software Processing yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino

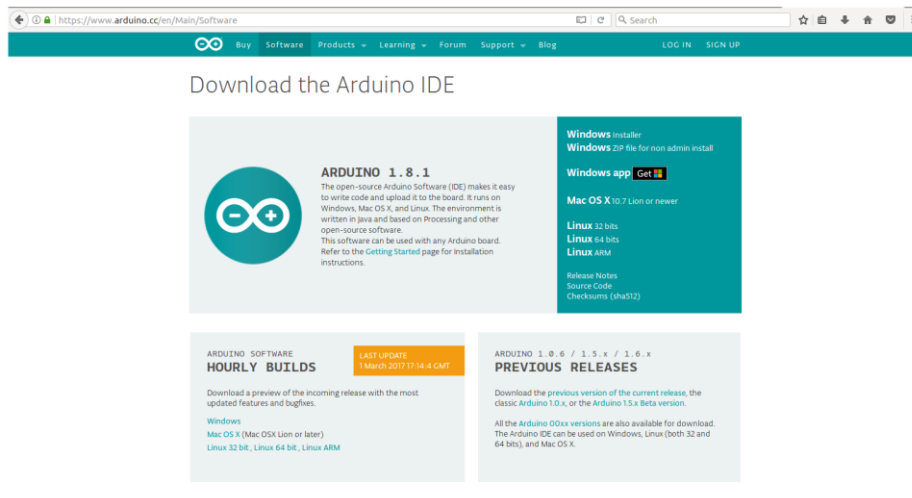


**Gambar 2.2** Software Arduino IDE [5]

### 2.5.1 Cara Menginstall Aplikasi Arduino IDE


Adapun langkah-langkah cara menginstall aplikasi Arduino IDE, yaitu:

Software Arduino IDE dapat di download di <https://www.arduino.cc> pilih **Windows Installer**, kemudian akan tampil pilihan Download, **Just Download** atau **Contribute and Download**.



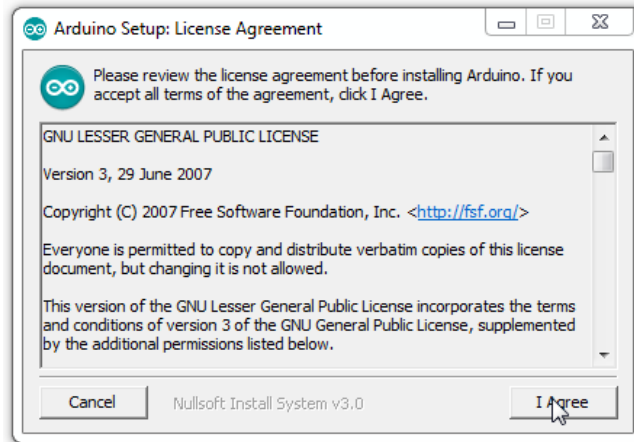
**Gambar 2.3** Laman Web Download Arduino IDE [5]

Buka aplikasi Arduino IDE dengan cara mengklik dua kali shortcut.

Name	Date modified	Type	Size
 arduino-1.8.12-windows.exe	4/17/2020 10:46 AM	Application	114,558 KB

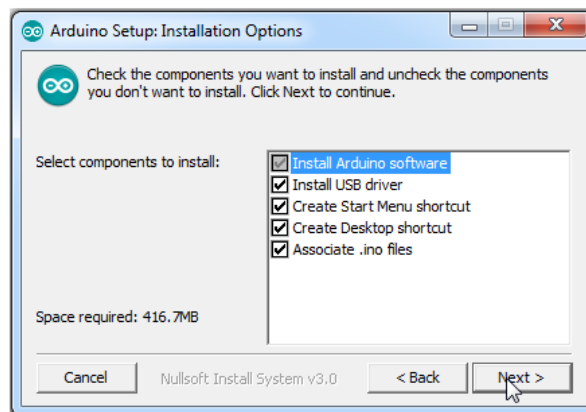
**Gambar 2.4** Aplikasi Installer Arduino IDE [5]

**License Agreement** atau **Persetujuan Instalasi**, klik tombol **I Agree** untuk memulai proses instalasi software.



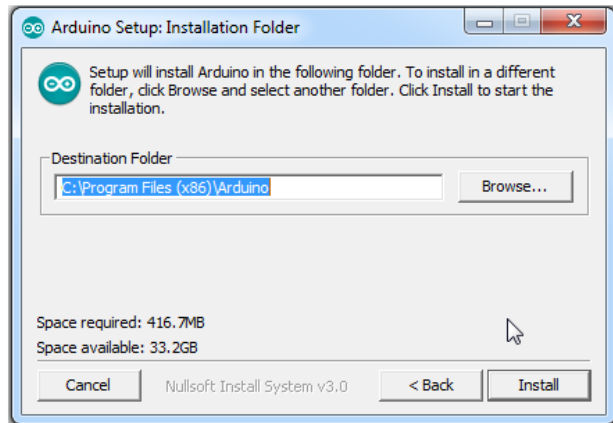
**Gambar 2.5** *Arduino License Agreement.* [5]

Untuk *Installation Option* pilih semua option dan klik tombol *Next*.



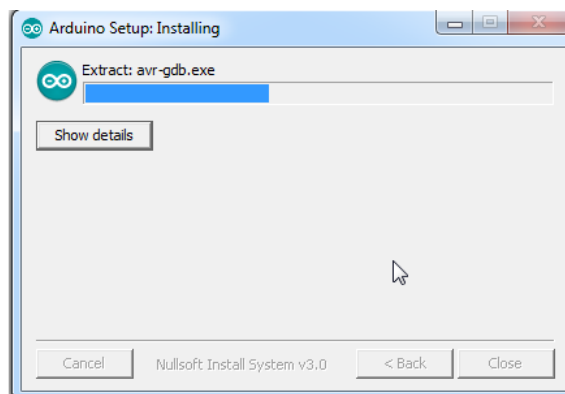
**Gambar 2.6** *Opsi Instalasi Arduinio IDE* [5]

**Installation Folder** untuk memilih folder tempat penyimpanan program arduino dan klik tombol **Install** untuk memulai proses instalasi *software*.



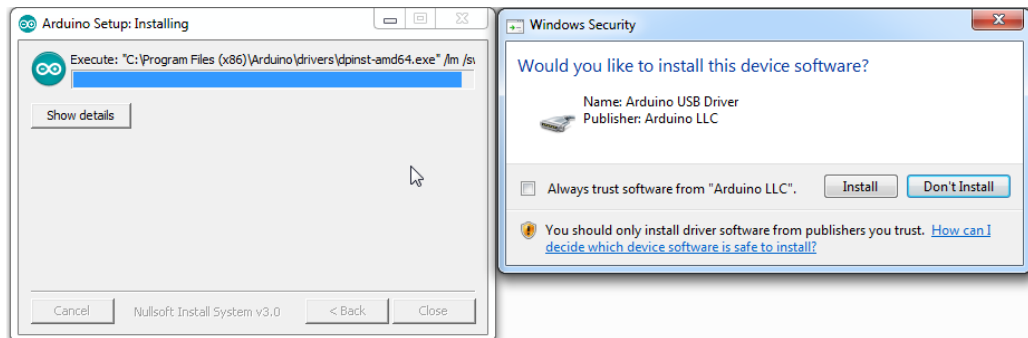
**Gambar 2.7** Folder Instalasi Arduino IDE [5]

Proses instalasi berlangsung, tunggu beberapa saat.



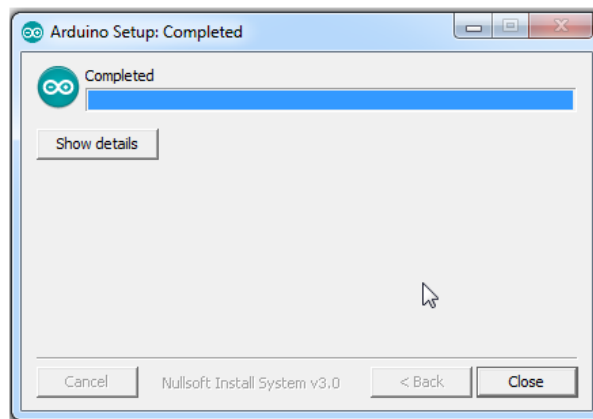
**Gambar 2.8** Proses Instalasi Arduino IDE [5]

saat proses instalasi berlangsung akan muncul pilihan untuk install driver, pilih **Install**, proses ini berguna untuk mengenali dan melakukan komunikasi dengan board arduino melalui port USB.



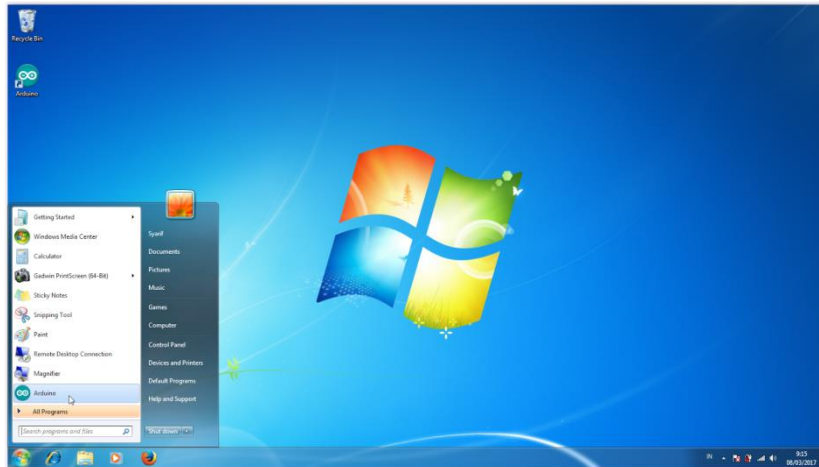
**Gambar 2.9** Proses Instalasi USB Driver [5]

Proses instalasi selesai



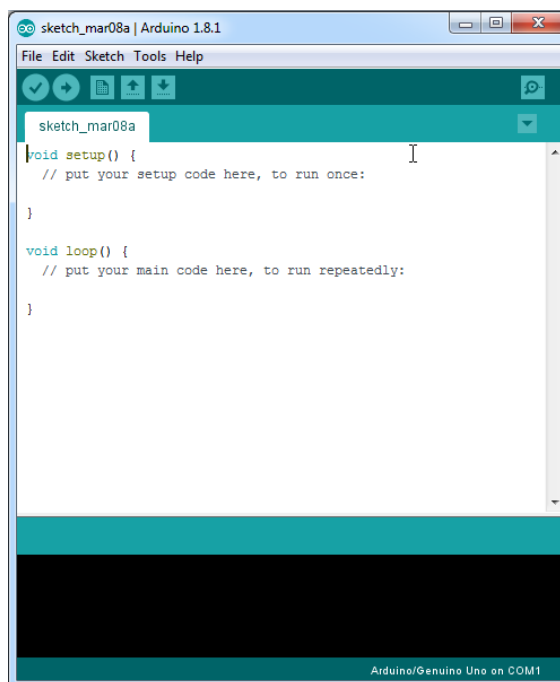
**Gambar 2.10** Proses Instalasi Arduino Selesai [5]

Proses instalasi selesai dan terinstall di windows. Cek di **Desktop Windows** atau **Start** atau **Start Menu** untuk menjalankan *Software* Arduino IDE. Double klik icon Arduino IDE di desktop atau klik di **Start Menu**.



**Gambar 2.11** Start Arduino IDE [5]







Setelah terbuka akan ada tampilan jendela *software* Arduino IDE, Proses instalasi Arduino IDE selesai.



**Gambar 2.12** Tampilan Jendela *Software* Arduino IDE [5]

Adapun bagian - bagian dalam *software* Arduino IDE, yaitu:

**Tabel 2.3** Menu Arduino IDE [5]

Simbol	Keterangan
	Verify Memeriksa kode Anda untuk kesalahan kompilasi itu.
	Upload Mengkompilasi kode Anda dan upload ke papan dikonfigurasi. Lihat upload di bawah ini untuk rincian.
	New Membuat sketsa baru.
	Open Membuka file yang sudah ada
	Save Mengamankan sketsa Anda.
	Serial Monitor Membuka Monitor serial .

### 2.5.2 Pemrograman Arduino

Pemrograman arduino menggunakan struktur Bahasa C. Mekanisme pemrograman arduino sama dengan mikrokontroler pada umumnya. Mulai dari membuat *sketch* program, meng-*compile*, selanjutnya proses *upload* pada papan arduino. Pengisian program dengan metode upload ialah mengisi papan arduino dengan program yang sudah berbentuk Hex atau hasil *compile* dari bahasa C ke bahasa mesin.

Program Arduino dapat dibagi dalam tiga bagian utama: struktur, nilai-nilai (variabel dan konstanta), dan fungsi. (Zamisyak Obi, 2017)

#### 1. Struktur Utama

##### a. Setup()

Fungsi `setup()` dipanggil ketika sketsa program dimulai. Fungsi ini digunakan untuk menginisialisasi variabel, mode pin, penggunaan librari, dll. Fungsi `setup()` hanya akan berjalan sekali, setelah power arduino dinyalakan atau saat mereset papan Arduino. Contoh : Program 1.1

```

int ledPin = 13;
void setup(){
pinMode(ledPin, OUTPUT);
digitalWrite(ledPin, HIGH);
delay(5000);
digitalWrite(ledPin, HIGH); }
void loop(){ // ... }

```

Program 1.1 akan menyalakan LED pada pin 13 selama 5 detik lalu mati. Eksekusi ini dilakukan hanya sekali.

### b. Loop()

Setelah membuat fungsi setup(), maka berikutnya adalah fungsi loop(). Fungsi loop() akan melakukan loop berturut-turut dimana program akan dijalankan terus menerus secara berurutan dan loop untuk mengontrol papan Arduino.

Contoh : Program 1.2

```

const int buttonPin = 3;
setup ()
{ Serial.begin(9600);
pinMode(buttonPin, INPUT); }
loop()
{ if (digitalRead (buttonPin) == HIGH)
Serial.write ( 'H');
else
Serial.write ( 'L');
delay (1000); }

```

Program 1.2 pada serial monitor akan menampilkan huruf H ketika tombol pada pin 3 ditekan dan bila dilepaskan akan tampil huruf L.

### c. //Komentar

Komentar digunakan untuk memberikan keterangan pada program yang dibuat. Komentar tidak dieksekusi maka komentar tidak menambah ukuran file hasil compile. Cara membuat komentar ialah sebagai berikut :



```
//komentar segaris diawali dengan dua garis miring
/*komentar untuk lebih dari satu baris diawali dengan garis miring lalu tanda bintang serta diakhiri dengan
bintang lalu garis miring*/
```

## 2. Ekspresi Bilangan

Dalam pemrograman bahasa C pada arduino, bilangan dapat diekspresikan dalam beberapa format, yaitu :

### a. Biner

Ditulis dengan awalan huruf '0b'. Contoh : b11110010

### b. Desimal

Ditulis biasa tanpa awalan. Contoh: 123

### c. Oktal

Ditulis dengan awalan angka "0". Contoh: 0345

### d. Heksadesimal

Diawali dengan "0x". Contoh: 0x5A

## 3. Struktur Kontrol

### a. if

Digunakan untuk mengecek suatu kondisi. Jika benar maka perintah didalam **if** akan dikerjakan.

if(kondisi){ Pernyataan / perintah }	Contoh : if(x==6) { a=a+5; }
---	---------------------------------

### b. if-else

Seperti dengan if, hanya saja ada 2 pilihan pernyataan / perintah. Jika kondisi benar maka perintah didalam if akan dikerjakan, jika kondisinya salah maka pernyataan didalam else lah yang akan dikerjakan

if(kondisi){ Pernyataan / perintah 1 } else { Pernyataan / perintah 2 }	Contoh : if(x==1) { a=1; } else { a=0; }
--	---

### c. if-else if

Untuk melakukan pengecekan suatu kondisi lebih dari satu maka bisa menggunakan if – else if.

if(kondisi1){ Pernyataan / perintah 1 } else if(kondisi2){ Pernyataan / perintah 2 } else if(kondisi ke-n){ Pernyataan / perintah ke-n }	Contoh : if(x==1) { a=1; } else if(x==2){ a=2; } else if(x==3){ a=3; }
---	---

### d. Switch case

Pernyataan ini digunakan untuk memilih kondisi yang sesuai untuk kemudian mengerjakan perintahnya. Bedanya adalah kondisi yang diuji berupa sebuah nilai variable. switch(variabel){ //variable yang diuji

case 1 : //pernyataan/perintah 1 break; case 2 : //pernyataan/perintah 2 break; case n : //pernyataan/perintah n break; default : //pernyataan/perintah default }	Contoh : switch(a){ case 1 : digitalWrite(pin1,HIGH) break; case 2 : digitalWrite(pin2,HIGH) break; case 3 : digitalWrite(pin3,HIGH) break; default : digitalWrite(pin4,LOW) }
---	---

Jika variable memenuhi syarat dari salah satu case maka dia akan mengerjakan pernyataan/perintah tersebut. Misal nilai variable = 2 maka dia kan mengerjakan pernyataan/perintah 2. Jika tidak memenuhi maka dia akan mengerjakan default.

## 4. Perulangan

### a. while

Perulangan ini digunakan untuk membuat perulangan yang tidak terbatas selama kondisi dalam while benar.

while(kondisi){ //pernyataan/perintah }	Contoh : while(a<200){ a++; }
--	----------------------------------

Perulangan while akan berhenti atau keluar setelah a mencapai angka 200.

### b. do ... while

Perulangan ini akan melakukan pernyataan /perintah lalu akan melihat kondisi dalam while. Jika benar maka pernyataan / perintah akan dieksekusi kembali.

do{ //pernyataan/perintah } while(kondisi);	Contoh : do{ a++; } while(a<200);
---	---

Perulangan penambahan a+1 akan dilakukan sampai nilai a=200.

### c. for

Digunakan untuk perulangan yang sifatnya terbatas.

for(inisialisasi;kondisi;step){ //pernyataan/perintah }	Contoh :or(a=0;a<=10;a++){ Serial.println(a); }
--	--

**Inisialisasi** : nilai awal suatu variabel untuk proses perulangan.

**Kondisi** : kondisi yang menentukan perulangan, jika benar pengulangan dikerjakan

**Step** : tahap perulangan bisa dalam bentuk perkalian, penambahan, pengurangan dan pembagian. Program tersebut akan menampilkan nilai a dari 0 sampai 10

#### **d. Goto**

Perintah ini digunakan untuk melompat/menuju perintah yang telah diberi label.

```
goto label;
Contoh : while(1){
digitalWrite(pin0,HIGH);
delay(1000);
digitalWrite(pin0,LOW);
delay(1000);
if(digitalRead(pin1)==HIGH);
{ goto keluar; } keluar:
```

#### **e. return**

Digunakan untuk memberikan nilai balik dari sebuah fungsi.

```
Contoh : int data(){
if(analogRead(A0)>100){
return 1;
else
return 0; }
```

#### **f. continue**

Untuk melewati perulangan yang tersisa dari struktur looping (do, for, atau while).

```
Contoh : for(a=0;a<=255;a+10){
if(digitalRead(pin0)==HIGH){
continue; }
digitalWrite(pwm1,a);
delay(100); }
```

#### **g. break**

Perintah 'keluar' dari pernyataan perulangan do, for, atau while. Juga digunakan untuk mengakhiri pernyataan dalam switch – case

## 5. Syntax

### a. semicolon

Digunakan untuk mengakhiri sebuah pernyataan.

### b. {} (*curly braces*)

Bagian utama dari bahasa pemrograman C yang digunakan dalam beberapa konstruksi yang berbeda dalam beberapa fungsi.

### c. #define

Komponen C yang berguna yang memungkinkan *programmer* untuk memberi nama untuk nilai konstan sebelum program dikompilasi.

### d. #include

Digunakan untuk memasukkan perpustakaan atau *library* di luar di sketsa program.

## 6. Operasi Aritmatika

Tabel 2.4 Operasi Aritmatika [5]

Operasi Aritmatika Operator	Keterangan
=	Pemberian Nilai
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Sisa Bagian

## 7. Operator Perbandingan

Tabel 2.5 Operator Perbandingan [5]

Operator	Keterangan
==	Persamaan. Jika kedua nilai yang dibandingkan sama maka hasilnya 'true'
!=	Pertidaksamaan. Jika kedua nilai yang dibandingkan tidak sama hasilnya 'true'
>	Lebih Besar
<	Lebih Kecil
>=	Lebih Besar atau Sama Dengan
<=	Lebih Kecil atau Sama Dengan

## 8. Operator Boolean

Tabel 2.6 Operator Boolean [5]

Operator	Keterangan
&&	AND
	OR
!	NOT

## 9. Operator Bitwise

Bitwise Operator = Digunakan untuk operasi bit per bit pada nilai integer. Terdiri dari operator NOT, AND, OR, XOR, Shl, Shr. Type : int atau char

Bitwise operator, dari namanya sudah jelas diketahui pasti berhubungan dengan bit. Biasanya digunakan utk memanipulasi data bertipe bit dari tipe data integer atau byte.

Tabel 2.7 Operator Bitwise [5]

Operator	Keterangan
<<	Geser Kiri
>>	Geser Kanan
&	AND
	OR
^	XOR
~	NOT

## 10. Operator Pertambahan dan Pengurangan

Tabel 2.8 Operator Pertambahan dan Pengurangan [5]

Operator	Keterangan	Contoh	Keterangan
++	Pertambahan 1 / <i>increment</i>	a++	a = a + 1
--	Pengurangan 1 / <i>decrement</i>	a--	a = a - 1

## 11. Variabel

Variabel adalah suatu wadah untuk menyimpan atau menampung data. Nama variable dibebaskan namun ada peraturan tersendiri seperti tidak boleh ada spasi, maksimal 32 karakter dan tidak boleh menggunakan istilah baku dalam bahasa C arduino karena dapat tersaji progam yang error. Cara mendeklarasikan variable sebelum digunakan yaitu sebagai berikut :

[tipe data][spasi][nama variabel][=][nilai int nilai\_1]

Contoh :int nilai\_1=17; //variabel bilangan tipe integer diisi nilai 17

## 12. Tipe Data

Tipe data yang berbeda – beda memiliki kapasitas penyimpanan yang berbeda – beda pula. Berikut tipe data tersebut :

**Tabel 2.9** Tipe-tipe Data [5]

<b>Tipe data</b>	<b>Lebar Data</b>	<b>Jangkauan</b>
Char	1 byte	-128 s/d 127
unsigned char	1 byte	0 s/d 255
Byte	1 byte	0 s/d 255
Word	2 byte	0 s/d 65535
Int	2 byte	-32768 s/d 32767
unsigned int	2 byte	0 s/d 65535
Long	4 byte	-2147438648 s/d 2147438647
unsigned long	4 byte	0 s/d 4294967295
Float	4 Byte	-3.4028235E+38 s/d 3.4028235E+38

## 13. Pin Input dan Output

Pada papan Arduino Uno terdapat 20 pin I/O yaitu 14 pin digital dan 6 pin analog.

### a. Inisialisasi Fungsi Pin I/O

Pada saat yang sama, sebuah pin hanya bisa memiliki satu fungsi saja baik input maupun output. Inisialisasi ini dilakukan pada fungsi setup(), dengan cara :

**pinMode(pin,mode)**

1. Pin : nomor pin yang dikonfigurasi dari papan arduino.
2. Mode : INPUT, INPUT\_PULLUP, OUTPUT

Sebagai Contohnya, jika pin no 3 akan dibuat menjadi Inputan maka : **pinMode(3,INPUT);** Bila pin 3 menjadi Outputan maka : **pinMode(3,OUTPUT);**

\*penulisan besar dan kecilnya huruf sangat berpengaruh. Perhatikan dengan seksama saat menulis program.

#### b. Menulis Data Digital di Pin Output

Setelah membuat pin sebagai digital output, selanjutnya untuk menulis atau mengeluarkan logika data digital dengan perintah sebagai berikut :

<b>digitalWrite(pin,value);</b>	Sebagai Contoh :
a. Pin : nomor pin digital output.	<b>pinMode(3,OUTPUT);</b>
b. Value : HIGH atau LOW.	<b>digitalWrite(3,HIGH);</b>

#### c. Membaca Data Digital pada Pin Input

Jika sebuah pin dibuat sebagai inputan maka kita harus menentukan aktif HIGH atau aktif LOW. Jika aktif HIGH maka dibutuhkan resistor pulldown. Jika memilih aktif LOW, cukup dengan memanggil resistor internal dengan pull up pada setiap pin arduino. Sebelum melakukan pembacaan maka perlu disetting untuk Inputannya. **pinMode(pin,mode)**

1. Pin : nomor pin yang dikonfigurasi dari papan arduino.
2. Mode : INPUT, INPUT\_PULLUP.

Setelah itu baru menuliskan ini : **digitalRead(pin);**

3. Pin : nomor pin arduino yang digunakan sebagai inputan.

Contoh : Pin 3 digunakan sebagai inputan pullup.

```
int baca;
pinMode(3,INPUT_PULLUP);
baca = digitalRead(3);
Serial.println(baca);
```

Hasil pembacaan pin 3 maka disimpan pada variabel baca.



#### d. Menulis Data Analog di Pin Output PWM

Untuk menggunakan `analogWrite()`, tidak perlu menggunakan `pinMode()` untuk mengatur pin sebagai output.

Cara menggunakannya sebagai berikut : **`analogWrite(pin,value);`**

1. Pin : nomor pin arduino yang digunakan sebagai outputan lihat table 1.1.

Contoh : Pin 5 digunakan sebagai outputan pwm. **`analogWrite(5,100);`**

2. Value : nilai pwm mulai dari 0-255.

#### e. Membaca Data Analog di Pin Input ADC (*Analog to Digital Converter*)

Untuk menggunakan `analogRead()`, tidak perlu menggunakan `pinMode()` untuk mengatur pin sebagai input.

Cara menggunakannya sebagai berikut : **`analogRead(analogPin);`**

`analogPin` : nomor pin arduino yang digunakan sebagai inputan analog (A0, A1, A2, A3, A4, A5).

Contoh : Membaca nilai analog pada `analogPin` A0 dan ditampilkan pada Serial.

```
int val = analogRead(A0); Serial.println(val);
```

### 14. Time

#### a. `millis()`

Menghitung dengan satuan miliseconds sejak papan Arduino mulai menjalankan program hingga 50 hari setelah itu akan kembali ke nol begitu selanjutnya

Contoh : **`unsigned long time = millis();`**  
**`Serial.println(time);`**

#### b. `micros()`

Menghitung dengan satuan microseconds sejak papan Arduino mulai menjalankan program hingga 70 menit setelah itu akan kembali ke nol begitu selanjutnya. Contoh : **`unsigned long time = micros();`**

```
Serial.println(time);
```

**c. delay()**

Jeda program untuk jumlah waktu (dalam milidetik). (Ada 1000 milidetik dalam satu detik). Contoh :

```
digitalWrite (ledPin, TINGGI); // set LED on
```

```
delay (1000); // menunggu untuk kedua
```

```
digitalWrite (ledPin, LOW); // set LED off
```

```
delay (1000); // menunggu untuk kedua
```

**d. delayMicrosecond()**

Jeda program untuk jumlah waktu (dalam mikrodetik). Ada seribu mikrodetik di milidetik, dan satu juta mikrodetik dalam detik. Contoh :

```
digitalWrite (ledPin, TINGGI); // set LED on
```

```
delayMicroseconds(50); // menunggu
```

```
digitalWrite (ledPin, LOW); // set LED off
```

```
delayMicroseconds(50); // menunggu
```

## **15. External Interrupts**

**a. attachInterrupt()**

Penulis harus menggunakan **digitalPinToInterrupt (pin)** untuk menerjemahkan pin digital sebenarnya untuk jumlah interrupt tertentu.

Misalnya, jika penulis terhubung ke pin 3, menggunakan **digitalPinToInterrupt (3)** sebagai parameter pertama yang attachInterrupt.

**Tabel 2.10** Digital Pins Usable for Interrupts pada attach Interrupt [5]

Board	Digital Pins Usable For Interrupts
Uno, Nano, Mini, other 328-based	2, 3
Mega, Mega2560, MegaADK	2, 3, 18, 19, 20, 21
Micro, Leonardo, other 32u4-based	0, 1, 2, 3, 7
Zero	all digital pins, except 4
MKR1000 Rev.1	0, 1, 4, 5, 6, 7, 8, 9, A1, A2
Due	all digital pins
101	all digital pins

Sintaknya adalah :

**Tabel 2.11** Sintak attachInterrupt [5]

attachInterrupt (digital Pin To Interrupt (pin), ISR, mode);	<i>(Disarankan)</i>
attach Interrupt (interrupt, ISR, mode);	<i>(Tidak disarankan)</i>
attachInterrupt (pin, ISR, mode);	<i>(Hanya Arduino Due, Zero, MKR1000,101 saja)</i>

Parameternya :

**Tabel 2.12** Parameter attachInterrupt [5]

interupsi:	jumlah interrupt ( <i>int</i> )	
pin:	nomor pin	<i>(ArduinoZero, MKR1000 saja)</i>
ISR:	ISR untuk panggilan ketika interupsi terjadi; fungsi ini harus ada parameter dan kembali apa-apa. Fungsi ini kadang-kadang disebut sebagai <i>rutin layanan interupsi</i> .	
Modus:	Interrupt harus dipicu. Empat konstanta yang telah ditetapkan sebagai nilai-nilai yang valid: 1. <b>LOW</b> untuk memicu interupsi setiap kali pin rendah,	

	<p>2. <b>CHANGE</b> memicu interupsi setiap kali pin perubahan nilai</p> <p>3. <b>RISING</b> untuk memicu ketika pin ganti dari rendah ke tinggi,</p> <p>4. <b>FALLING</b> ketika pin ganti dari tinggi ke rendah.</p>	
	<p>The papan Arduino Due memungkinkan untuk:</p> <p><b>HIGH</b> untuk memicu interupsi setiap kali pin yang tinggi.</p>	( <i>ArduinoZero, MKR1000 saja</i> )

Contoh : **const byte ledPin = 13;**

**const byte interruptPin = 2;**

**volatile byte state = LOW;**

**void setup() {**

pinMode(ledPin, OUTPUT);

pinMode(interruptPin, INPUT\_PULLUP);

**attachInterrupt(digitalPinToInterrupt(interruptPin),                    blink,**  
**CHANGE); } void loop() {**

digitalWrite(ledPin, state); }

**void blink() {**

state = !state; }

**b. detachInterrupt()**

Mematikan interupsi yang diberikan.

Sintaks :

**Tabel 2.13** Sintak detachInterrupt [5]

detachInterrupt ( <i>interrupt</i> )	
detachInterrupt (digitalPinToInterrupt (pin));	
detachInterrupt ( <i>pin</i> )	( <i>Arduino Karena, Nol saja</i> )

## 16. Interrupts

### a. interrupts()

Mengaktifkan kembali interupsi (setelah dinonaktifkan oleh `noInterrupts()`). Interupsi memungkinkan melakukan tugas-tugas penting tertentu di belakang program utama dan diaktifkan secara default. Beberapa fungsi tidak akan bekerja saat interupsi dinonaktifkan, dan komunikasi yang masuk dapat diabaikan.

Contoh : `void setup() {}`

```
void loop(){
  noInterrupts(); // critical, time-sensitive code here
  interrupts(); // other code here }
```

### b. noInterrupts

Menonaktifkan interupsi (dapat mengaktifkan kembali mereka dengan interupsi `()`). Interupsi memungkinkan melakukan tugas-tugas penting tertentu di belakang program utama dan diaktifkan secara default.

Contoh : `void setup() {}`

```
void loop(){
  noInterrupts();
  // critical, time-sensitive code here
  interrupts();
  // other code here }
```

## 17. Communication

### a. Serial

Komunikasi serial merupakan komunikasi dua arah dari Transmitter dengan Receiver dan sebaliknya. Kita bisa melakukan komunikasi serial dengan memanfaatkan pin Rx dan Tx pada arduino maupun pada USB. Cara menggunakan serial sebagai berikut ,

```
Contoh : void setup() {  
    Serial.begin(9600); }  
  
void loop(){  
    unsigned char a = a++;  
    Serial.println(a); }
```

## 2.6 Aplikasi Blynk

Blynk adalah aplikasi IoT Cloud untuk aplikasi iOS dan Android yang bertujuan sebagai kontrol untuk modul Arduino, Raspberry Pi, ESP8266, Wemos D1, dan modul sejenisnya menggunakan koneksi internet. Aplikasi ini merupakan *platform* antarmuka grafis untuk proyek yang di implementasikan dengan metode *drag and drop widget*. Pengoperasian Blynk sangat mudah dan sederhana, membuat alat siap terhubung untuk Internet of Things. Blynk tidak terikat pada papan atau modul tertentu, Dari *platform* aplikasi ini dapat mengontrol benda apapun dari jarak jauh, dimanapun berada dan dalam waktu *real time* dengan catatan terhubung dengan internet dan koneksi internet yang stabil. Blynk diciptakan dengan tujuan untuk kontrol dan monitoring hardware secara jarak jauh menggunakan komunikasi data internet ataupun intranet (jaringan LAN). Kemampuan untuk menyimpan data dan menampilkan data secara visual baik menggunakan angka, warna ataupun grafis semakin memudahkan penulis dalam proses pembuatan proyek dibidang Internet of Things. (Mahali,2017)



**Gambar 2.13** Aplikasi Blynk [7]