



### 2.1.1 Contoh Aplikasi IoT

Banyak sekali contoh penerapan teknologi IoT, diantaranya yaitu:

1. *Smart Home*, sistem keamanan rumah berbasis internet yang dapat mengetahui keadaan rumah serta mengontrol peralatan rumah tangga melalui jaringan internet.
2. *Smart City*, implementasi *Smart City* dilakukan dengan mengintegrasikan sarana *public* seperti lampu jalan, tempat pembuangan sampah sementara, hingga sensor parkir dengan internet.
3. Internet Industri, contohnya *monitoring* dan kontroling peralatan serta proses *industry*.
4. Kesehatan, *monitoring* kondisi kesehatan seseorang.
5. Transportasi, manajemen dan informasi lalu lintas.

### 2.1.2 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 2.2 Skema IoT

(Sumber : [www.mobnasesemka.com](http://www.mobnasesemka.com))

## 2.2 Arduino

Menurut website resmi Arduino, Arduino merupakan sebuah perangkat elektronik yang bersifat *open source* dan sering digunakan untuk merancang dan membuat perangkat elektronik serta *software* yang mudah untuk digunakan. Arduino ini dirancang sedemikian rupa untuk mempermudah penggunaan perangkat elektronik di berbagai bidang.<sup>[4]</sup>

Arduino memiliki beberapa komponen penting di dalamnya, seperti pin, mikrokontroler, dan konektor yang nanti akan dibahas lebih dalam selanjutnya. Selain itu, Arduino juga sudah menggunakan bahasa pemrograman *Arduino Language* yang sedikit mirip dengan bahasa pemrograman C++.<sup>[4]</sup>

Biasanya Arduino digunakan untuk mengembangkan beberapa sistem seperti pengatur suhu, sensor untuk bidang agrikultur, pengendali peralatan pintar, dan masih banyak lagi.<sup>[4]</sup>

### Komponen Arduino

Arduino memiliki beberapa komponen yang penting di dalamnya. Berikut ini adalah penjelasan dari masing-masing komponen Arduino :

#### 1. Mikrokontroler

Mikrokontroler adalah chip yang memungkinkan untuk memprogram Arduino dan memproses output berdasarkan *input* yang diberikan. Singkatnya, mikrokontroler ini adalah otak dari Arduino. Ada banyak jenis *chip* yang digunakan tergantung dari jenis Arduino-nya.

#### 2. Pin

Pin ini digunakan untuk menghubungkan Arduino dengan berbagai komponen yang akan digunakan. Dalam Arduino sendiri ada dua jenis pin, yakni pin analog dan pin digital.

##### a. Pin digital

Pin ini dapat menerima atau mengirim sinyal digital. Digital berarti sinyal yang diterima atau dikirimkan akan bernilai 1 atau 0 alias *HIGH* atau *LOW*. Kebanyakan perangkat Arduino memiliki 14 pin *input output* digital.

#### b. Pin digital

Pin analog pada arduino adalah pin yang digunakan untuk menerima *input* analog. Pin ini dapat menerima tegangan analog dari 0V sampai dengan 5V. Umumnya, setiap jenis Arduino memiliki setidaknya satu pin analog.

Setiap pin pada Arduino biasanya dapat dikonfigurasi ke dalam dua mode, yaitu *input* dan *output*. Pada mode *input*, pin akan diatur untuk dapat menerima sinyal *input*. Sama halnya pada mode *output*, pin akan diatur untuk mengirimkan sinyal.<sup>[4]</sup>

### 3. Konektor

Arduino sendiri memiliki dua jenis konektor yang cukup penting, yaitu *power* konektor dan serial konektor.

#### a. Power konektor

Power konektor adalah konektor yang digunakan untuk menyalurkan daya untuk Arduino. Daya ini digunakan untuk menghidupkan Arduino dan juga perangkat lain yang terhubung dengannya, seperti sensor dan layar *monitoring*.

#### b. Serial konektor

Serial konektor ini biasanya digunakan untuk menghubungkan Arduino dengan perangkat seperti komputer atau laptop. Konektor ini menggunakan *port* USB standar pada Arduino. Selain itu, konektor ini juga dapat digunakan sebagai *power* konektor. Namun, serial konektor hanya diimplementasikan pada perangkat Arduino yang lebih baru.

## **Kelebihan dan Kekurangan Arduino**

Jika kita membahas perangkat mikrokontroler, Arduino memiliki daya tarik tersendiri bagi para penggunanya. Pasalnya, ia memiliki beberapa kelebihan yang membuatnya semakin bersaing dengan mikrokontroler lainnya. Berikut ini adalah kelebihan dan kekurangannya :

### **1. Kelebihan**

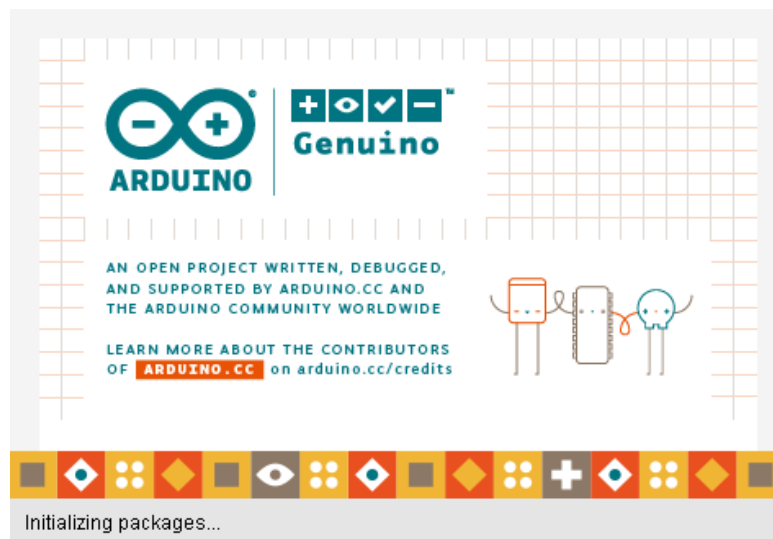
- a. Modul Arduino yang tersedia sudah siap digunakan (*shield*) seperti modul GPS, LAN, dan SD *card reader*.
- b. Harga yang relatif terjangkau.
- c. Mudah digunakan oleh pemula.
- d. Ada banyak *library* yang dapat digunakan untuk memudahkan kamu dalam bereksperimen.
- e. Memiliki *port* USB yang dapat digunakan untuk transfer data dan untuk mengalirkan sumber daya.
- f. Memiliki Arduino IDE yang digunakan untuk menulis dan meng-upload program ke mikrokontroler.
- g. Konsumsi daya yang rendah.

### **2. Kekurangan**

- a. Tidak bisa diinstal OS sehingga tidak dapat digunakan sebagai komputer pribadi.
- b. Jika kamu ingin mengubah atau memodifikasi program lama, kamu harus memodifikasi seluruh program.
- c. Beberapa tipe Arduino tidak menyediakan modul *wired* atau *wireless* secara *built-in*.
- d. Memiliki kapasitas memori yang kecil.
- e. Ruang penyimpanan terpotong karena digunakan untuk *bootloader*.
- f. Memiliki *clock speed* yang rendah.

### 2.2.1 Arduino IDE

Arduino IDE adalah *software* yang digunakan untuk membuat *sketch* pemrograman atau dengan kata lain arduino IDE sebagai media untuk pemrograman pada *board* yang ingin diprogram. IDE adalah kependekan dari *Integrated Development Environment*. Arduino IDE ini berguna untuk mengedit, membuat, meng-upload ke *board* yang ditentukan, dan meng-*coding* program tertentu. Arduino IDE dibuat dari bahasa pemrograman JAVA, yang dilengkapi dengan *library* C/C++(*wiring*), yang membuat operasi *input/output* lebih mudah. *Software* arduino IDE ini tidak hanya untuk memprogram *board* arduino UNO, tetapi juga untuk memprogram *board* yang lainnya seperti arduino nano, arduino genio, mappi32, nodeMCU, dan sejenisnya.<sup>[5]</sup>









Gambar 2.3 Tampilan Aplikasi Arduino IDE

(Sumber : [www.kmtech.id](http://www.kmtech.id))

*Arduino Integrated Development Environment* atau *Arduino Software (IDE)* merupakan *software* yang digunakan sebagai media untuk memprogram di arduino, dengan kata lain Arduino IDE sebagai media untuk memprogram *board* Arduino. Arduino IDE ini berguna sebagai *text editor* untuk membuat, mengedit, dan juga mevalidasi kode program. Bisa juga digunakan untuk meng-*upload* ke *board* Arduino. Kode program yang digunakan pada Arduino disebut dengan istilah Arduino “*sketch*” atau disebut juga *source code* arduino, dengan ekstensi file *source code*.<sup>[5]</sup>

Tabel 2.1 *Menu* Arduino IDE

Simbol	Keterangan
	<i>Verify</i> Berfungsi untuk melakukan <i>checking</i> kode yang kamu buat apakah sudah sesuai dengan kaidah pemrograman yang ada atau belum.
	<i>Upload</i> Berfungsi untuk melakukan kompilasi program atau kode yang kamu buat menjadi bahasa yang dapat dipahami oleh <i>mesh</i> alias si Arduino.
	<i>New Sketch</i> Berfungsi membuat <i>sketch</i> baru.
	<i>Open</i> Membuka <i>file</i> yang sudah ada.
	<i>Save</i> Berfungsi untuk menyimpan <i>sketch</i> yang telah kamu buat.
	Serial Monitor Berfungsi untuk membuka serial monitor. Serial monitor disini merupakan jendela yang menampilkan data apa saja yang dikirimkan atau dipertukarkan antara arduino dengan <i>sketch</i> pada <i>port</i> serialnya. Serial monitor ini dapat digunakan untuk menampilkan nilai proses, nilai pembacaan, bahkan pesan <i>error</i> .

### 2.2.2 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.<sup>[5]</sup>

Program Arduino dapat dibagi dalam tiga bagian utama: Struktur, nilai-nilai (variabel dan konstanta), dan fungsi [ ].

#### 1. Struktur utama

##### a. Setup()

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

##### 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.

##### c. //Komentar

Komentar digunakan untuk memberikan keterangan pada program yang dibuat. Komentar ini tidak dieksekusi maka komentar tidak menambah ukuran *file* hasil *compile*.



## 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 : 350

### c. Oktal

Ditulis dengan awalan angka '0'. Contoh : 0550

### d. Heksadesimal

Diawali dengan '0x'. contoh : 0x5A

## 3. Struktur Kontrol

Setiap program yang dibuat membutuhkan suatu kontrol. Tak hanya perulangan namun suatu eksekusi dengan syarat tertentu juga diperlukan. Pengujian kondisi sebagai berikut :

### a. *if*

Digunakan untuk mengecek suatu kondisi, jika benar maka perintah didalam *if* akan dieksekusi.

### b. *if - else*

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

### c. *if - else if*

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

### d. *Switch case*

Pernyataan ini digunakan untuk memilih kondisi yang sesuai untuk kemudian mengerjakan perintahnya. Bedanya adalah

kondisi yang diuji berupa sebuah nilai variabel, *switch* (variabel)  
{// variabel yang diuji.

Jika variabel memenuhi syarat dari salah satu *case* maka dia akan mengerjakan pernyataan/perintah tersebut. Misal nilai variabel = 2 maka dia akan mengerjakan pernyataan/perintah 2. Jika tidak memenuhi maka secara otomatis yang akan dieksekusi adalah *default*.

#### **4. Perulangan**

##### **a. While ()**

Perulangan ini digunakan untuk membuat perulangan yang tidak terbatas selama kondisi didalamnya terpenuhi. Program akan terus melakukan perulangan hingga nilai a mencapai angka 100.

##### **b. do...while()**

Perintah ini berfungsi untuk mengeksekusi suatu pernyataan atau perintah lalu melihat kondisi di dalam *while*. Apabila kondisinya sesuai maka pernyataan akan dieksekusi kembali. Dan jika tidak maka pernyataan tak akan dieksekusi. Perulangan penambahan a+1 akan dilakukan sampai nilai a = 100.

##### **c. For ()**

Fungsi *for* yaitu untuk melakukan perulangan yang sifatnya terbatas. Dalam perintah *for* wajib disertakan nilai awal, kondisi, dan penambahan (*increment*) atau pengurangan (*decrement*).

##### **d. goto**

Perintah ini digunakan untuk melompat secara langsung atau menuju perintah yang telah diberi label.

##### **e. return**

Untuk memberikan nilai balik dari suatu fungsi.

##### **f. continue**

Digunakan untuk melewati perulangan yang tersisa dari struktur *looping*. Baik itu *do*, *for*, maupun *while*.

**g. break**

Digunakan untuk memaksa sebuah perulangan berhenti sebelum waktunya. Baik itu dalam perulangan *for*, *while*, maupun *do..while*.

**5. Syntax**

Dalam bahasa pemrograman, *syntax* diartikan sebagai suatu aturan dalam penulisan kode program agar kode tersebut bisa dimengerti komputer. Beberapa *syntax* dalam pemrograman arduino antara lain:

**a. Titik koma ; (*semicolon*)**

Digunakan untuk mengakhiri sebuah pernyataan.

**b. Kurung kurawal {} (*curly braces*)**

Bagian utama dalam 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 memasukan perpustakaan atau *library* di luar di sketsa program.

## 6. Operasi Aritmatika

Tabel 2.2 Operasi Aritmatika

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

## 7. Operator Perbandingan

Tabel 2.3 Operator Perbandingan

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 2.4 Operator Boolean

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, XORShl, Shr. Type : int atau char. *Bitwise* operator, dari namanya sudah jelas diketahui pasti berhubungan dengan bit. Biasanya digunakan untuk memanipulasi data bertipe bit dari tipe data integer atau *byte*.

Tabel 2.5 Operator *Bitwise*

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

## 10. Operator Pertambahan dan Pengurangan

Tabel 2.6 Operator Pertambahan dan Pengurangan

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 variabel 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 program yang *error*. Cara mendeklarasikan *variable* sebelum digunakan yaitu sebagai berikut : [tipe data][spasi][nama variabel][=][nilai int nilai\_1];.

## 12. Tipe Data

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

Tabel 2.7 Tipe-tipe Data

<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. Time

### a. Milis()

Menghitung dengan satuan *millisecond* sejak papan arduino mulai menjalankan program hingga 50 hari setelah itu akan kembali ke nol begitu selanjutnya.

### b. Micros()

Menghitung dengan satuan *microseconds* sejak papan arduino mulai menjalankan program hingga 70 menit setelah itu akan kembali ke nol.

### c. Delay

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

### d. Delay Microseconds()

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

### 2.3 WiFi (*Wireless Fidelity*)

WiFi merupakan salah satu aplikasi pengembangan *wireless* untuk komunikasi data. Sesuai dengan namanya yaitu *wireless*, berarti tanpa kabel. WiFi adalah jaringan lokal yang tidak menggunakan kabel. WiFi dirancang berdasarkan spesifikasi IEEE 802.11 n yang berada pada frekuensi 2.4GHz dengan data *rate* 100Mb/s. Sebuah alat yang dapat memakai WiFi (seperti komputer pribadi, telepon pintar, tablet, atau pemutar audio digital) dapat terhubung dengan sumber jaringan seperti internet melalui sebuah titik akses jaringan nirkabel. Titik akses (atau hotspot) seperti itu mempunyai jangkauan sekitar 20 meter (65 kaki) di dalam ruangan dan lebih luas lagi di luar ruangan. Cakupan hotspot dapat mencakup wilayah seluas kamar dengan dinding yang memblokir gelombang radio atau beberapa mil persegi, ini bisa dilakukan dengan memakai beberapa titik akses yang saling tumpang tindih. Jaringan WiFi sangat efektif digunakan didalam sebuah kawasan atau gedung. Dengan performa dan keamanan yang dapat diandalkan, pengembangan jaringan WiFi menjadi trend baru pengembangan jaringan menggantikan jaringan *wire* atau jaringan penuh kabel.<sup>[6]</sup>

#### 2.3.1 Spesifikasi WiFi

IEEE 802.11 adalah spesifikasi yang terdiri dari 4 standarisasi yaitu untuk mengimplementasikan komunikasi komputer *wireless local area network* di frekuensi 2.4, 3.6, 5, dan 60 GHz. Diciptakan dan dioperasikan oleh *Institute of Electrical and Electronics Engineers*. Versi dasar dirilis tahun 1997 dan telah melalui serangkaian pembaruan dan menyediakan dasar bagi produk jaringan nirkabel WiFi.<sup>[6]</sup>

Tabel 2.8 Spesifikasi *Wireless Fidelity*

Spesifikasi	Kecepatan	Frekuensi Band
<b>802.11 b</b>	11 Mb/s	2.4GHz
<b>802.11 a</b>	54 Mb/s	5GHz
<b>802.11 g</b>	54 Mb/s	2.4GHz
<b>802.11 n</b>	100Mb/s	2.4GHz

1. Standarisasi IEEE 802.11 b

Standar 802.11 b saat ini yang paling banyak digunakan. Menawarkan *throughput* maksimum dari 11 Mbps (6 Mbps dalam praktik) dan jangkauan hingga 300 meter di lingkungan terbuka, menggunakan rentang frekuensi 2,4 GHz, dengan 3 saluran radio yang tersedia. Transmisi data 5,4 hingga 11 Mbps.

2. Standarisasi IEEE 802.11 a

Standard IEEE 802.11 a bekerja pada frekuensi 5 GHz mengikuti standar dari UNII (*Unlicensed National Information Infrastructure*). Teknologi IEEE 802.11 a tidak menggunakan teknologi *spread-spectrum* melainkan menggunakan standar *frequency division multiplexing* (FDM) dan mampu mentransfer data hingga 54 Mbps.

3. Standarisasi IEEE 802.11 g

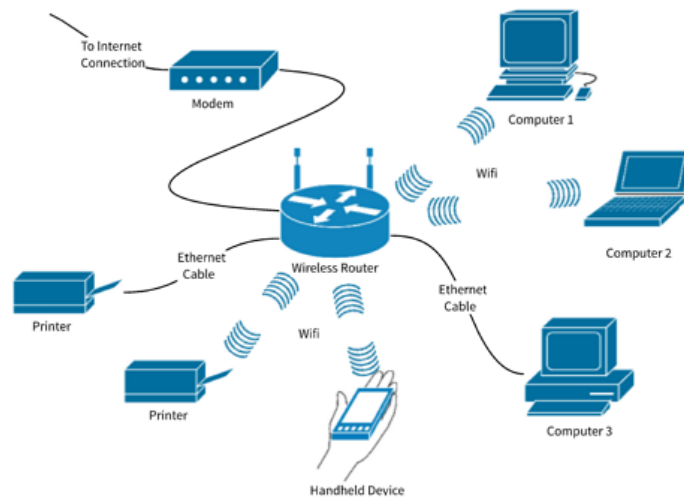
Standar 802.11 g menawarkan *bandwidth* yang tinggi (54 Mbps *throughput* maksimum, 30 Mbps dalam praktik) pada rentang frekuensi 2,4 GHz. Standar 802.11 g mundur kompatibel (mampu bekerja dan bergerak dengan serasi/sama) dengan standar 802.11 b, yang berarti bahwa perangkat yang mendukung standar 802.11 g juga dapat bekerja dengan 802.11 b.

4. Standarisasi IEEE 802.11 n

Spesifikasi IEEE 802.11 n yang berada pada *frekuensi* 2.4GHz dengan data *rate* 100Mb/s. Jaringan WiFi sangat efektif digunakan didalam sebuah kawasan atau gedung. Dengan performa dan keamanan yang dapat diandalkan, pengembangan



jaringan WiFi menjadi trend baru pengembangan jaringan menggantikan jaringan *wire* atau jaringan penuh kabel.



Gambar 2.4 WiFi

(Sumber : [www.hipwee.com](http://www.hipwee.com))

### 2.3.2 Fungsi Wifi

Berikut berbagai macam kegunaan Wifi selain digunakan untuk internetan dari berbagai sumber :

#### a. Mensinkronisasikan telepon dengan PC tanpa USB

Wifi bisa digunakan untuk sinkronisasi hp dengan laptop/komputer. Sehingga memudahkan untuk tidak repot menggunakan kabel data ataupun kabel USB hanya dengan melalui aplikasi.

#### b. Mentransfer foto dari kamera digital

Kartu Eye-fi adalah kartu memori tanpa nirkabel dan saat kamera sedang dalam jangkauan jaringan tertentu, foto dan video akan tertransfer ke komputer / laptop.

#### c. Streaming film ke TV

Media server dapat streaming video melalui wifi untuk setiap HTPC lainnya. Xbox/ Playstation-diaktifkan TV di rumah.

### 2.3.3 Mode Akses Koneksi Wifi

Dalam Wifi, terdapat 2 mode akses koneksi yaitu :

a. Ad-Hoc

Mode koneksi ini adalah mode dimana beberapa komputer terhubung secara langsung, atau lebih dikenal dengan istilah *peer-to-peer*. Keuntungannya, lebih murah dan praktis bila yang terkoneksi hanya 2 atau 3 komputer, tanpa harus membeli *access point* milik Anto Peo.

b. Infrastruktur

Menggunakan *access point* yang berfungsi sebagai pengatur lalu lintas data, sehingga memungkinkan banyak *client* dapat saling terhubung melalui jaringan.

#### **Aplikasi *Massachusetts Institute of Technology* (MIT)**

Aplikasi ini adalah sistem berbasis *web* dimana aplikasi *android* dapat digunakan tanpa perlu tahu bagaimana cara meng-*code*-nya. Sistem ini telah dihentikan oleh *Google* tapi dirilis kembali oleh *Google* sebagai proyek *open-source* dan saat ini dikelola oleh *Massachusetts Institute of Technology* (MIT). Dengan *app inventor*, pengguna bisa melakukan pemrograman komputer untuk menciptakan aplikasi perangkat lunak dengan sistem operasi berbasis *android*. *App inventor* ini berbasis *visual block programming* karena memungkinkan pengguna bisa menggunakan, melihat, menyusun dan men-*drag and drops block* yang merupakan simbol perintah dan fungsi *event handler* untuk menciptakan sebuah aplikasi yang bisa berjalan di sistem *android*.<sup>[7]</sup>

MIT *App Inventor* merupakan *platform* untuk memudahkan proses pembuatan aplikasi sederhana tanpa harus mempelajari atau menggunakan bahasa pemrograman yang terlalu banyak. Kita dapat mendesain aplikasi *android* sesuai keinginan dengan menggunakan berbagai macam *layout* dan komponen yang tersedia. *App Inventor* memungkinkan pengguna baru untuk memprogram komputer untuk menciptakan aplikasi perangkat lunak bagi sistem operasi *Android*. *App Inventor* menggunakan antarmuka grafis, serupa dengan antarmuka pengguna pada *Scratch* dan StarLogo TNG, yang memungkinkan pengguna untuk men-*drag-*

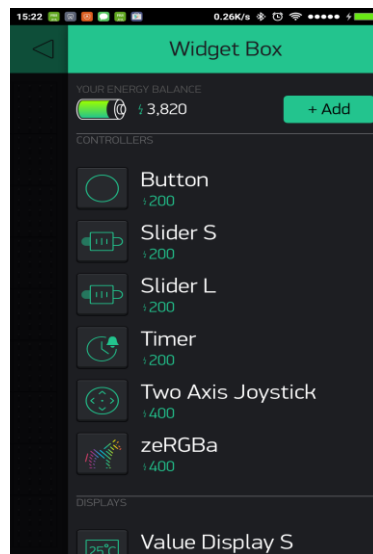
*and-drop* objek visual untuk menciptakan aplikasi yang bisa dijalankan pada perangkat *android*. Dalam menciptakan *App Inventor*, *Google* telah melakukan riset yang berhubungan dengan komputasi edukasional dan menyelesaikan lingkungan pengembangan *online Google*.<sup>[7]</sup>



Gambar 2.5 MIT *App Inventor*  
(Sumber : [psti.unisayogya.ac.id](http://psti.unisayogya.ac.id))

#### 2.4 Aplikasi *Blynk*

*Blynk* adalah aplikasi untuk iOS dan OS *Android* untuk mengontrol *Arduino*, *NodeMCU*, *Raspberry Pi* dan sejenisnya melalui internet. Aplikasi ini dapat digunakan untuk mengendalikan perangkat *hardware*, menampilkan data sensor, menyimpan data, visualisasi, dan lain-lain. Aplikasi *Blynk* memiliki 3 komponen utama, yaitu Aplikasi, Server, dan *Libraries*. *Blynk* server berfungsi untuk menangani semua komunikasi diantara smartphone dan *hardware*. *Widget* yang tersedia pada *Blynk* diantaranya adalah *Button*, *Value Display*, *History Graph*, *Twitter*, dan Email. *Blynk* tidak terikat dengan beberapa jenis mikrokontroler namun harus didukung *hardware* yang dipilih. *NodeMCU* dikontrol dengan internet melalui *WiFi*, chip *ESP8266*. *Blynk* akan dibuat online dan siap untuk *Internet of Things*.<sup>[8]</sup>



Gambar 2.6 Aplikasi *Blynk*  
(Sumber : bagi2ilmuaditya.blogspot.com)

## 2.5 *Android*

*Android* merupakan salah satu sistem operasi atau *operating system* berbasis *mobile* yang sangat banyak di gunakan sekarang ini. Utamanya pada telepon pintar (*smartphone*) ataupun tablet.<sup>[9]</sup>

*Android* adalah sistem operasi yang dirancang oleh *Google* dengan basis kernel *Linux* untuk mendukung kinerja perangkat elektronik layar sentuh, seperti tablet atau *smartphone*. Jadi, *android* digunakan dengan sentuhan, gesekan ataupun ketukan pada layar *gadget* anda. *Android* bersifat *open source* atau bebas digunakan, dimodifikasi, diperbaiki dan didistribusikan oleh para pembuat ataupun pengembang perangkat lunak. Dengan sifat *open source* perusahaan teknologi bebas menggunakan OS ini di perangkatnya tanpa lisensi alias gratis.<sup>[9]</sup>

Di awal pembuatannya, *android* ditargetkan bagi penggunaan perangkat kamera digital. Akan tetapi, para pencipta *android*, yaitu Andy Rubin, Chris White, dan Nick Sears berpendapat bahwa pasar untuk kamera digital tidak terlalu besar. Maka dari itu, sistem operasi ini kemudian dialihkan penggunaannya pada ponsel pintar.<sup>[9]</sup>

Pada tahun 2004, *android* mulai dipasarkan dan berhadapan dengan saingan *smartphone* berbasis sistem operasi *Symbian* dan *Windows Mobile*. Di awal pemasarannya ini, Andy Rubin dan partner-nya sulit mendapatkan investor. Hingga akhirnya, *android* berhasil mendapatkan suntikan dana sebesar 10.000 dolar Amerika dari Steve Perlman, seseorang yang kala itu ingin membantu Andy Rubin. Di bulan Juli 2005, *Google* mengakuisisi *Android Inc.* dengan uang sebesar 50 juta dolar. Para pendiri *android* kemudian bergabung dengan *Google* dan memimpin proyek ini. Setelah *Google* akhirnya berkompetisi juga dalam perangkat ponsel pintar yang dibelinya, yaitu *android*, *Google* akhirnya membuat prototipe. Prototipe tersebut merupakan *smartphone* yang memiliki keyboard, seperti milik *Blackberry*, Hingga Desember 2006, berita mengenai prototipe *Android* ini terus tersiar. Tanpa disangka-sangka, pada tahun 2007, perusahaan *Apple* merilis *iPhone* dengan desain *smartphone* yang hampir seluruh permukaannya menggunakan layar sentuh. Mulai dari situ, *Google* memikirkan bagaimana perkembangan *smartphone android* untuk ke depannya, mengingat prototipe awalnya menggunakan keyboard tanpa layar sentuh sama sekali. Untuk menyaingi *iPhone*, *Nokia* dan *Blackberry* merilis ponsel dengan layar sentuh di tahun 2008. Tak ingin kalah dengan kompetitornya, *Google* juga merilis ponsel dengan layar sentuh, yaitu *HTC Dream* atau *T-Mobile G1*. Namun, tak hanya layar sentuh saja, *smartphone* ini juga tidak meninggalkan penggunaan keyboard.<sup>[9]</sup>

### **Kelebihan *Android***

#### **1. Merupakan Sistem Operasi *Open Source***

Siapa saja bisa menggunakannya secara gratis. Para *developer* atau pengembang dimudahkan untuk mengoptimalkan dan mengembangkan OS ini untuk *smartphone* yang dibuatnya.

#### **2. Harganya Beragam**

Ada yang terbilang cukup terjangkau, ada pula yang memiliki harga jual tinggi. Sehingga, *smartphone android* bisa menjangkau semua kalangan. Namun, semakin tinggi harga, semakin mumpuni pula spesifikasinya.

### 3. Memiliki Banyak Dukungan Aplikasi

Hal ini juga tidak lepas dari sifat *android* yang merupakan sistem operasi *Open Source*. Pengembang pun diizinkan untuk mengembangkan aplikasi berbasis *source code* dan *android*. Oleh karena itu, jika anda masuk ke *Play Store*, akan ditemukan banyak sekali ribuan aplikasi yang sesuai dengan kebutuhan pengguna.

### 4. Mudah dimodifikasi

Banyak komponen yang bisa anda atur ulang atau dimodifikasi, mulai dari ROM hingga *custom overclock* pada sistem operasi. Hal ini bisa berpengaruh terhadap performa ponsel pintar berbasis *android* agar bisa bekerja lebih cepat dan sesuai dengan keinginan.

## **Kekurangan Android**

### 1. Kerja sistemnya cukup berat

Hal ini menyebabkan banyak memori yang dibutuhkan, baik RAM maupun ROM. Bagi *smartphone* yang memiliki RAM dan ROM berkapasitas kecil, tentu ini akan menghambat performanya.

### 2. Hasil modifikasi sering menyebabkan sistem bekerja tidak stabil dan kurang optimal. Ada kalanya hasil modifikasi mengakibatkan OS menjadi sedikit lelet dan kurang responsif. Nantinya bisa berpengaruh pada *hardware* sehingga menjadi cepat panas dan kapasitas memori lebih mudah bocor.