

BAB II

TINJAUAN PUSTAKA

2.1. Perbandingan Penelitian Sejenis

Seperti yang telah dipaparkan pada latar belakang bahwa penulis menggunakan beberapa jurnal yang sejenis pada penelitian ini sebagai pembanding juga referensi. Disini penulis membandingkan tiga jurnal yang diambil dari sisi keunggulan juga kelemahan masing-masing jurnal, untuk keterangan lebih lanjut bisa dibaca pada tabel dibawah ini.

Tabel 2.1 Tabel Perbandingan Peneliti yang Sejenis

NO	Judul	Nama Peneliti	Teknologi Yang Dipakai	Keunggulan	Kelemahan
1.	Sistem Buka Tutup Pintu Otomatis Berbasis Suara Manusia (2018)	Sinta Ariyanti, Slamet Seno Adi, Sugeng Purbawanto	Sensor Suara, <i>Easy VR</i> , Mikrokontroler Atmega328	Pada sistem ini, pengenalan suaranya menggunakan modul Easy VR. Pengujian pada alat yang dilakukan oleh orang yang berwenang (suara telah terekam pada program) besar presentase tingkat	Pemberian perintah oleh orang lain rata – rata presentase keberhasilannya adalah 13% untuk kata buka dan 4% untuk kata tutup.

				keberhasilannya adalah 95% untuk kata buka dan 90% untuk kata tutup.	
2.	Rancang Bangun Buka Tutup Pintu Otomatis Menggunakan Pengenalan Isyarat Tutar (2017)	A. Asni B, Aswadul Fitri Saiful Rahman, Muzakky Mursyid	Modul Easy VR, Sensor Mikrofon, Modul Arduino	Modul EasyVR memiliki tingkat keberhasilan penerimaan suara sebesar 8,5% jika menerima perintah suara dari orang yang berbeda. Dan dapat menerima suara sebesar 90% untuk orang yang sama dengan melakukan sampling suara.	Masih perlu pengembangan lebih lanjut untuk mengatasi kekurangan di-antaranya bahasa yang dikenali oleh modul masih menggunakan bahasa Inggris, dan masih rentan terhadap gangguan derau.
3.	Sistem Pintu Cerdas Menggunakan Sensor	A. Farha Adella, Muh. Fardika	Sensor Ultrasonik, ESP-32, Motor	Jika seseorang dalam radar sensor ultrasonik	Belum tersedianya pengamanan yang lebih

	Ultrasonic berbasis <i>Internet of Things</i> (2020)	Pratama Putra, Farros Taufiqurrahman, Andi Baso Kaswar	Servo	berjarak 8cm, dan bantuan motor servo dengan posisi 140 derajat membutuhkan delay 4s maka pintu akan terbuka secara otomatis.	mendetail sehingga hanya dapat diakses oleh orang tertentu.
4.	Aplikasi Pengaman Pintu Lemari Penyimpanan di Bengkel PCB Politeknik Negeri Sriwijaya dengan SIM900A berbasis Android (2021)	Cenika Mutiara, Fitriyani Astuti	Arduino Uno, <i>Bluetooth</i> HC-06, SIM900A	Akan membuka kunci pintu lemari menggunakan keypad dan akan memberikan notifikasi berupa sms apabila ada yang membuka lemari secara paksa.	Belum ada

2.2. Android

2.2.1. Pengertian Android

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 diperangkatnya tanpa lisensi alias *gratis* [1]. Begitupun dengan para pembuat aplikasi, mereka bebas membuat aplikasi dengan kode-kode sumber yang dikeluarkan google. Dengan seperti itu android memiliki jutaan *support* aplikasi gratis/berbayar yang dapat diunduh melalui google play.

Seperti software lain, Android memiliki nomor versi. Selain nomor versi, Google juga menyematkan nama julukan untuk setiap versinya yang berbasiskan nama makanan penutup. Jika anda sudah lama mengenal Android, tentu hal ini bukanlah hal aneh.



Gambar 2.1. Logo Android^[1]

2.2.2. Sejarah Android

Android adalah sistem operasi yang berbasis *Linux* untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan *platform* terbuka

bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak [2]. Pada bulan Juli 2005, Google mencetuskan kerjasama dengan Android Inc. yang saat itu berlokasi di California, Amerika Serikat. Kala itu para pendiri Android.Inc beranggapan bahwa sistem operasi Android hanya diperuntukkan pada telepon seluler. Sehingga munculah isu bahwa Google hendak memasuki pasar telepon seluler dengan menggandeng Android sebagai sistem operasinya.

Akhirnya pada periode September 2007, Google memperkenalkan Nexus One, salah satu jenis smartphone GSM dengan sistem operasi berbasis Android. Google juga mengajukan hak paten atas aplikasi pada smartphone ini dan kemudian smartphone ini diproduksi oleh HTC Corporation dan mulai dipasarkan pada Januari 2010. Selain itu, pada September 2008 terbentuklah anggota-anggota baru yang bergabung dalam program kerja Android ARM Holdings, yakni Sony Ericsson, Toshiba Corp, SoftBank, Vodafone Group dan beberapa perusahaan lainnya [3].

2.2.3. Perkembangan Tipe Android Dari Waktu ke Waktu

Sejak pertama kali diluncurkan hingga sekarang, Android senantiasa melakukan perbaruan melalui perbaikan bug dan penambahan fitur-fitur baru. HTC Dream yang dirilis pada Oktober 2008 merupakan smartphone pertama yang menggunakan sistem operasi berbasis Android. Hingga saat ini tak terhitung lagi berapa banyak smartphone yang mengusung Android sebagai sistem operasi.

Evaluasi dan pengembangan yang tiada henti menjadi salah satu kunci kesuksesan Android dalam merebut hati para pengguna smartphone dan gadget. Salah satu ciri khas Android adalah penamaan tipe-tipenya berdasarkan urutan abjad dan nama-nama makanan. Hal ini pula yang membuat para pengguna smartphone dan gadget mudah mengingat tipe-tipe Android apa saja yang telah diluncurkan. Berikut ini adalah ulasan singkat mengenai perkembangan tipe Android dari waktu ke waktu:

1. Android Alpha dan Android Beta

Sistem Android yang dikenal dengan nama Android Alpha dan Android Beta ini pertama kali diperkenalkan pada tahun 2007 dan baru mulai diaplikasikan pada smartphone di awal Maret 2009. Sebagai cikal bakal OS Android, versi ini bisa dikatakan cukup berhasil menjadi inisiator kesuksesan besar Android.

2. Android 1.5 (Cupcake)

Sesaat setelah mengaplikasikan sistem operasinya pada smartphone, pada Mei 2009 Android kembali merilis versi terbarunya yang diberi nama Android Cupcake. Android Cupcake menawarkan beragam kelebihan dibandingkan dengan versi terdahulunya, yakni fasilitas unggah video ke Youtube, headset bluetooth yang nirkabel serta tampilan keyboard dan gambar yang lebih atraktif.

3. Android 1.6 (Donut)

Tipe Android yang satu ini diluncurkan hanya berselang 4 bulan setelah peluncuran saudara kandungnya, Android Cupcake. Android Donut telah mengusung keunggulan lainnya yakni tampilan indikator baterai, fasilitas zoom in zoom out, penggunaan koneksi CDMA serta beberapa keunggulan lainnya.

4. Android 2.0 / 2.1 (Eclair)

Masih di tahun 2009, Android kembali meluncurkan teknologi terbarunya yang diberi nama Android Éclair. Era Android Éclair kemudian berhasil menarik para perusahaan gadget untuk mulai membuat gadget dengan sistem layar sentuh yang sebelumnya dianggap kurang user friendly bagi para pengguna smartphone.

5. Android 2.2 (Froyo = Frozen Yoghurt)

Android Froyo diluncurkan pada Mei 2010, 5 bulan setelah peluncuran Android Éclair. Pada tipe Android yang satu ini, keinginan para pengguna smartphone untuk memiliki kapasitas memori eksternal berupa slot micro SD sudah dapat diwujudkan.

6. Android 2.3 (Ginger Bread)

Si roti jahe yang diluncurkan pada penghujung tahun 2010 memiliki tampilan yang atraktif dan disertai dengan penambahan fitur-fitur seperti dual camera dan video call. Tak hanya itu saja, Ginger Bread juga fokus pada peningkatan kualitas dan grafis game berbasis Android.

7. Android 3.0/3.1 (Honeycomb)

Android 3.0/3.1 (Honeycomb) yang diluncurkan pada Mei 2011 ini dikhususkan bagi para pengguna tablet PC berbasis Android. User interface nya pun berbeda dengan smartphone Android. Spesifikasi hardware yang tinggi serta tampilan layar yang lebih besar membuat Honeycomb cocok diaplikasikan pada tablet PC.

8. Android 4.0 (Ice Cream Sandwich)

Ice Cream Sandwich diluncurkan pada tahun yang sama dengan peluncuran Honeycomb. Banyak sekali fitur-fitur baru yang disematkan pada Android versi empat ini, diantaranya adalah fitur pendeteksi wajah, fitur yang memaksimalkan kualitas fotografi, kualitas cideo yang lebih baik serta resolusi dan grafis gambar yang sangat memuaskan.

9. Android 4.1 (Jelly Bean)

Sistem Android yang satu ini memberikan support pada fitur on screen keyboard yang membuat kegiatan mengetik menjadi lebih cepat, mudah dan responsif. Salah satu smartphone keren yang mengusung Android Jelly Bean adalah Google Nexus 7 yang diprakarsai oleh perusahaan ASUS.

10. Android 4.4 (Kitkat)

Android Kitkat merupakan versi Android terbaru yang diluncurkan pada bulan September 2013. Penggunaan nama Kitkat menjadi suatu kejutan bagi para pecinta Android, karena nama Kitkat merupakan salah satu nama snack wafer yang populer di dunia. Penggunaan nama ini juga membuat tipe Android ini semakin mudah diingat orang.

11. Android v5.0 – 5.1 Lollipop

Dirilis pada tanggal 15 Oktober 2014, versi OS ini mengusung perubahan besar dari segi UI yang nampak lebih flat dengan konsep material design. Versi Android ini sudah mendukung arsitektur 64-bit sehingga sudah memungkinkan untuk penggunaan RAM diatas 3 GB pada hardware perangkat. Penggunaan prosesor 64-bit pun makin banyak diadopsi oleh para vendor, mulai dari penerapan pada perangkat flagship hingga perangkat kelas menengah kebawah.

12. Android v6.0 Marshmallow

Versi Android ini resmi dirilis pada bulan September tahun 2015. Bersamaan dengan dirilisnya versi ini, untuk pertama kalinya Google juga memperkenalkan 2 perangkat smartphone Nexus sekaligus yang diproduksi oleh 2 vendor yang berbeda.

13. Android v7.0 Nougat

Resmi diperkenalkan pada akhir Juni 2016. Banyak netizen yang berspekulasi bahwa kemungkinan besar, pemberian nama untuk Android versi “N” ini adalah Nutella. Namun Google menepis kabar tersebut setelah resmi memperkenalkannya bersamaan dengan dipamerkannya patung icon Android yang berdiri diatas potongan Nougat (yang sepiintas lebih mirip dengan tempe itu).

2.3. Software

2.3.1. Pengertian *Software*

Menurut Roger S. Pressman (2002) definisi *software* (perangkat lunak) merupakan suatu perintah program dalam sebuah komputer yang apabila diperintahkan untuk dijalankan oleh *user* atau pengguna, mereka akan memberikan fungsi dan bekerja seperti yang diminta.

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”

2.3.2. Jenis-Jenis *Software*

1. *System Software*

Jenis software yang pertama adalah perangkat lunak sistem atau *system software*. Ini adalah perangkat lunak yang berfungsi sebagai dasar untuk perangkat lunak aplikasi. *System software* ini mencakup driver perangkat komputer, OS (sistem operasi), kompiler, pemformat disk, *teks_editor*, dan utilitas lain yang membantu komputer beroperasi lebih efisien. Jenis *system software* ini juga bertanggung jawab untuk mengelola komponen *hardware* (perangkat keras) dan menyediakan fungsi dasar non-tugas khusus. Perangkat lunak sistem biasanya ditulis dalam bahasa pemrograman C.

2. *Programming Software*

Berikutnya adalah jenis perangkat lunak pemrograman atau yang dikenal dengan *programming software*. Ini adalah seperangkat alat untuk membantu pengembang dalam menulis program. Berbagai alat yang tersedia adalah *compiler* (kompiler), *connector*, penerjemah dan editor teks.

3. *Application Software*

Perangkat lunak aplikasi atau *application software* adalah jenis dan kategori software yang dimaksudkan untuk melakukan tugas-tugas tertentu. Contoh dari *application software* ini termasuk seperti Office Suite, aplikasi game, sistem database, perangkat lunak pendidikan dan lain sebagainya. *Programming software* dapat berupa program tunggal atau kumpulan program kecil. Jenis kategori perangkat lunak ini adalah apa yang biasanya dianggap konsumen dengan sebutan “software”.

2.4 Bluetooth

2.4.1. Pengertian Bluetooth

Bluetooth adalah sebuah teknologi komunikasi wireless (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 GHz unlicensed ISM (Industrial, Scientific and Medical) dengan menggunakan sebuah frequency hopping transceiver yang mampu menyediakan layanan komunikasi data dan suara secara real-time antara host-host bluetooth dengan jarak jangkauan layanan yang terbatas. Bluetooth sendiri dapat berupa card yang bentuk dan fungsinya hampir sama dengan card yang digunakan untuk wireless local area network (WLAN) yang menggunakan frekuensi radio standar IEEE 802.11, hanya saja pada bluetooth mempunyai jangkauan jarak layanan yang lebih pendek dan kemampuan transfer data yang lebih rendah [4].

2.4.2. Cara Kerja Bluetooth

Bluetooth bekerja menggunakan frekuensi radio. Beda dengan infra merah yang mendasarkan diri pada gelombang cahaya. Jaringan Bluetooth bekerja pada frekuensi 2.402 Giga Hertz sampai dengan 2.480 Giga Hertz, dibangkitkan dengan daya listrik kecil sehingga membatasi daya jangkauannya hanya sampai 10 meter. Penetapan frekuensi ini telah distandardisasi secara internasional untuk peralatan elektronik yang dipakai untuk kepentingan industri, ilmiah, dan medis.

Kecepatan transfer data Bluetooth rilis 1.0 adalah 1 mega bit per detik (Mbps), sedangkan versi 2.0 mampu menangani pertukaran data hingga 3 Mbps.

2.5 Arduino IDE







Arduino Integrated Development Environment (IDE) adalah aplikasi lintas platform (untuk Windows, macOS, Linux) yang ditulis dalam fungsi dari C dan C ++. Ini digunakan untuk menulis dan mengunggah program ke papan yang kompatibel dengan Arduino, tetapi juga, dengan bantuan inti pihak ketiga, papan pengembangan vendor lainnya.

Kode sumber untuk IDE dirilis di bawah GNU General Public License, versi 2. Arduino IDE mendukung bahasa C dan C ++ menggunakan aturan khusus penataan kode. Arduino IDE memasok pustaka perangkat lunak dari proyek Wiring, yang menyediakan banyak prosedur input dan output umum. Kode yang ditulis pengguna hanya memerlukan dua fungsi dasar, untuk memulai sketsa dan loop program utama, yang dikompilasi dan dihubungkan dengan program stub main () ke dalam program eksekutif siklik yang dapat dieksekusi dengan rantai alat GNU, juga disertakan dengan distribusi IDE. Arduino IDE menggunakan program avrdude untuk mengubah kode yang dapat dieksekusi menjadi file teks dalam pengkodean heksadesimal yang dimuat ke papan Arduino oleh program pemuat di papan firmware.

Pembahasan kali ini, akan mengulas lebih detail bagian-bagian dari Arduino IDE seperti fungsi dari tombol-tombol yang terdapat pada aplikasi Arduino IDE. Mengulas tentang bagian-bagian dari aplikasi ini merupakan hal penting untuk dapat mempermudah dalam mengoperasikan sebuah aplikasi. Seperti aplikasi teks editor pada umumnya yaitu memiliki fitur untuk *cut / paste* dan untuk *find / replace* teks. Pada bagian keterangan aplikasi memberikan pesan balik saat menyimpan dan mengeksport dan juga sebagai tempat menampilkan kesalahan. Konsol log menampilkan output teks dari Arduino Software (IDE), termasuk pesan kesalahan yang lengkap dan informasi lainnya. Pojok kanan bawah jendela menampilkan papan dikonfigurasi dan port serial. Tombol toolbar memungkinkan

Anda untuk memverifikasi dan meng-upload program, membuat, membuka, dan menyimpan sketch, dan membuka monitor serial.

Tabel 2.2 Menu Arduino IDE

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 .

2.5.1. Pemrograman Arduino

Pemrograman arduino menggunakan struktur Bahasa C. Mekanisme pemrogramanya arduino sama dengan mikrokontroler pada umumnya. Mulai dari membuat sket progam, meng-compile,selanjutnya proses upload pada papan arduino. Pengisian progam dengan metode upload ialah mengisi papan arduino

dengan program yang sudah berbentuk Hex atau hasil compile dari bahasa C ke bahasa mesin [5].

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 :

1. Biner

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

2. Desimal

Ditulis biasa tanpa awalan. Contoh : 435

3. Oktal

Ditulis dengan awalan angka '0'. Contoh : 0753

4. 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 dikerjakan.

if(kondisi){	Contoh : if(x==6) {
Pernyataan / perintah }	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){	Contoh : if(x==1) {
Pernyataan / perintah 1 }	a=1; }
else {	else {
Pernyataan / perintah 2 }	a=0; }

c. if – else if

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

if(kondisi1){	Contoh : if(x==1) {
Pernyataan / perintah 1 }	a=1; }
else if(kondisi2){	else if(x==2){
Pernyataan / perintah 2 }	a=2; }
else if(kondisi ke-n){	else if(x==3){
Pernyataan / perintah ke-n }	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	Contoh : switch(a){
break;	case 1 : digitalWrite(pin1,HIGH)
case 2 : //pernyataan/perintah 2	break;

break;	case 2 : digitalWrite(pin2,HIGH)
case n : //pernyataan/perintah n	break;
break;	case 3 : digitalWrite(pin3,HIGH)
default : //pernyataan/perintah default }	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){	Contoh : while(a<200){
//pernyataan/perintah }	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{	Contoh : do{
//pernyataan/perintah }	a++; }

```
while(kondisi);           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){ Contoh :or(a=0;a<=10;a++){
//pernyataan/perintah } Serial.println(a); }
```

Inisialisasi : nilai awal suatu variable untuk proses perulangan.

Kondisi : kondisi yang menentukan proses perulangan, jika benar perulangan 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; }
digitaWrite(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 2.3 Operasi Aritmetika

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

7. Operator Perbandingan

Tabel 2.4 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 2.5 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, 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 2.6 Operator Bitwise

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

10. Operator Pertambahan dan Pengurangan

Tabel 2.7 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 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 variable][=][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 2.8 Tipe-tipe Data

Tipe data	Lebar Data	Jangkauan
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.

1. 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)`

- a. Pin : nomor pin yang dikonfigurasi dari papan arduino.
- b. 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.

2. Menulis Data Digital di Pin Output

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

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

3. 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)`**

- a. Pin : nomor pin yang dikonfigurasi dari papan arduino.
- b. Mode : INPUT, INPUT_PULLUP.

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

- c. 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.

4. 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);**

- a. Pin : nomor pin arduino yang digunakan sebagai outputan lihat table 1.1.
Contoh : Pin 5 digunakan sebagai outputan pwm. **analogWrite(5,100);**
- b. Value : nilai pwm mulai dari 0-255.

5. 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);**

- a. 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. delayMicroseconds()

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

1. attach Interrupt()

Anda harus menggunakan `digitalPinToInterrupt (pin)` untuk menerjemahkan pin digital sebenarnya untuk jumlah interrupt tertentu.

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

Tabel 2.9 Digital Pins Usable for Interrupts pada `attach Interrupt`

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 2.10 Sintak attachInterrupt

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 2.11 Parameter attachInterrupt

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: <ol style="list-style-type: none"> 1. LOW untuk memicu interupsi setiap kali pin rendah, 2. CHANGE memicu interupsi setiap kali pin perubahan nilai 3. RISING untuk memicu ketika pin ganti dari rendah ke tinggi, 	

	4. FALLING ketika pin ganti dari tinggi ke rendah.	
	The papan Arduino Due memungkinkan untuk: HIGH untuk memicu interupsi setiap kali pin yang tinggi.	<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; }

2. **detachInterrupt()**

Mematikan interupsi yang diberikan.

Sintak :

Tabel 2.12 Sintak detachInterrupt

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

16. Interrupts

1. 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 }

```

2. 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

1. 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 MIT App Inventor

MIT App Inventor adalah platform online yang dirancang untuk mengajarkan konsep berpikir komputasional melalui pengembangan aplikasi seluler. Membuat aplikasi dengan cara *dragg* dan *dropp* komponen ke tampilan desain dan menggunakan bahasa blok visual untuk memprogram perilaku aplikasi

MIT App Inventor merupakan 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 [6].

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.



Gambar 2.2 Logo MIT App Inventor ^[6]