

## **Arduino Internet of Things (IoT) Training**

**RDD** Technologies



#### Universitas Islam Bandung (UNISBA) Aug 2020



T: +62 811540593 M: support@rdd-tech.com W



#### **Major Components of IoT**









#### **Pengenalan Microcontroller**

#### 1. Definisi mikrokontroler

Mikrokontroler adalah sebuah komputer yang dilengkapi dengan semua komponen pendukung yang terpasang bersamanya (*on-board*) Mikrokontroler didesain untuk melakukan satu tugas dan menjalan program tertentu dengan konsumsi daya rendah. Pada *chip* mikrokontroler umumnya telah terdapat RAM, ROM (EEPROM, EPROM, PROM), I/O, sirkuit logika yang memiliki fungsi spesifik seperti UART, I2C, SPI dan *clock generator*.

(D. Calcutt, F. Cowan dan H. Prachizadeh, 8051 Microcontrollers, Burlington: Elsevier, 2004)



#### **Pengenalan Mikrokontroler**

Berikut dibawah ini merupakan diagram blok dari sebuah mikrokontroler:





#### Pengenalan Mikrokontroler

Mikrokontroler memiliki banyak jenis, tipe, arsitektur, dan merk, contohnya seperti ATmega328P yang berarsitektur AVR yang diproduksi ATmel, STM32F4L0 yang berarsitektur ARM cortex m4 yang diproduksi STmicroelectronic, dan lain-lain.



- Arduino merupakan platform prototyping elektronika yang bersifat open-source yang berbasis pada hardware dan software yang fleksibel dan mudah digunakan. Arduino diciptakan untuk digunakan oleh desainer (developer), pehobi elektronika dan semua orang yang tertarik dalam membuat objek interaktif atau dalam mengembangkan sebuah alat.
- Arduino memiliki banyak jenis misalnya Arduino Uno, Nano, due, Mega.
- Perbedaan jenis-jenis Arduino tersebut terletak pada jenis mikrokontroler yang dipakai, mulai lebar data (bit); 8bit hingga 32 Bit dan kecepatan *clock* 16MHz hingga diatas 100MHz.



- Arduino bekerja selayaknya komputer, bekerja sesuai dengan program yang dipasang dan dapat diprogram ulang (*reprogramable*).
- Pada sektor industri Arduino bisa digunakan/diibaratkan seperti PLC.
- Arduino membuat bahasa peemrograman yang mudah dipahami oleh semua orang, bahkan oleh anak-anak.
- Dikarenakan Arduino merupakan open-source platform. Komunitas pengguna Arduino sangat banyak, sehingga memudahkan untuk berbagi program dan troubleshooting.



## **Pengenalan Arduino**

- Program dibuat menggunakan komputer (PC) dengan menggunakan aplikasi "Arduino IDE", aplikasi ini gratis dan bisa didownload dari website Arduino.
- Setelah program dibuat di PC, kemudian program diunggah atau *upload* ke Arduino dengan menggunakan kabel USB.





**RDD** Technologies



Arduino Uno (Generasi Pertama Arduino, 8 bit 16Mhz Clock)



Arduino Mega (8Bit I/O lebih banyak) Arduino MKR1000 (32 Bit Dengan Wifi)



#### 2. Pengenalan perangkat keras Arduino MKR1000

MKR1000 merupakan sebuah *board* arduino yang didesain untuk Internet of Things project, Di dalamnya sudah tertanam module WiFi WINC1500 low power 2.4GHz. Mikrokontroler yang di pakai pada *board* ini adalah SAMD21 Cortez-M0 32bit yang di kembangkan oleh Atmel. MKR 1000 memiliki port untuk melakukan pengisian batre Li-Po. Port tersebut juga dapat digunakan sebagai sumber tegangan untuk menghidupkan MKR1000.



#### Berikut dibawah ini merupakan bentuk *board* dari Arduino MKR1000:





Berikut dibawah ini merupakan spesifikasi dari Arduino MKR1000:

Mikrokontroler	SAMD21 Cortex-M0+ 32bit Low Power ARM
Tegangan masukan	5VDC
Supported battery	Li-Po single cell, 3.7V, 700mAh minimum
Tegangan operasi	3.3VDC
Digital I/O	8 Pin
PWM	12 Pin
UART	1 Channel
SPI	1 Channel
12C	1 Channel
Analog input	7 Pin



Analog output	1 Pin
External Interrupts	8 Pin
Arus maksimum per I/O	7mA (DC)
Kapasitas memori flash	256KB
Kapasitas SRAM	32KB
Clock Speed	32.768 kHz(RTC), 48 MHz
Dimensi (PxL)	61.5mm x 25mm
Berat	32gr



#### **Contoh Aplikasi Arduino**







#### **Pengenalan Arduino IDE**

#### 1. Materi singkat mengenai Arduino IDE

Arduino IDE merupakan *software* khusus untuk memprogram *board* Arduino. Sintaks pemograman pada Arduino IDE menggunakan bahasa C yang telah dimodifikasi sedemikian rupa agar memudahkan dalam proses *coding*.

#### 2. Instalasi Arduino IDE

- a) Unduh *software* Arduino IDE pada *website* resmi Arduino.
- b) Buka *software* Arduino IDE yang telah diunduh tersebut sampai muncul jendela seperti gambar di bawah ini, lalu tekan tombol "*I Agree*".



🥺 Arduino S	etup: License Agreement	_		×
Please accep	e review the license agreement before t all terms of the agreement, click I Ag	installing Ardu ree.	ino. If you	
SNU LESSER	GENERAL PUBLIC LICENSE			^
Version 3, 29	June 2007			
Copyright (C)	2007 Free Software Foundation, Inc.	. < <u>http://fsf.or</u>	<u>'a/</u> >	
Everyone is p document, bu	ermitted to copy and distribute verbain at changing it is not allowed.	tim copies of th	is license	
This version of and condition by the addition	of the GNU Lesser General Public Licens s of version 3 of the GNU General Pub anal permissions listed below.	se incorporates lic License, sup	s the terms plemented	~
Cancel	Nullsoft Install System v3.0		I Agre	e



c) Setelah tombol "*I Agreed*" ditekan maka akan muncul jendela seperti gambar di bawah, lalu ceklis semua list yang ada pada bagian "*Select component to install*:", setelah itu tekan tombol "*next*".

🥺 Arduino Setup: Installation	Options	_		×
Check the components y you don't want to install.	ou want to install ar Click Next to contir	nd uncheck the nue.	e compone	ents
Select components to install:	<ul> <li>✓ Install Arduit</li> <li>✓ Install USB d</li> <li>✓ Create Start</li> <li>✓ Create Desk</li> <li>✓ Associate .in</li> </ul>	no software river : Menu shortcu top shortcut no files	ıt	
Space required: 539.6MB				
Cancel Nullsoft Insta	II System v3.0	< Back	Nex	t >



d) Setelah tombol "*Next*" ditekan maka akan muncul jendela seperti gambar dibawah ini, lalu pilih *directory* tempat untuk menyimpan Arduino IDE. Klik tombol "*Install*" untuk memulai proses instalasi.

💿 Arduino Setup	: Installation Folder	_		$\times$
Setup will i folder, did installation	nstall Arduino in the following fol k Browse and select another fold	der. To install ler. Click Instal	in a differer I to start th	nt e
Destination Fold	er es (x86)\Arduino		Browse	
,		_		
Space required: 53	39.6MB			
Space available: 1	83.8GB			
Cancel	Nullsoft Install System v3.0	< <u>B</u> ack	Inst	all



e) Setelah proses instalasi selesai, tekan tombol "close".





f) Buka Arduino IDE yang telah di-*install* tersebut, jika Arduino IDE berhasil dibuka maka akan muncul tampilan tatap muka (*interface*) seperti gambar dibawah ini.

💿 sketch_jan06a   Arduino 1.8.10		_		×
Eile Edit Sketch Tools Help				
				<b>1</b> 20-
sketch_jan06a				
<pre>void setup() {     // put your setup code here, to run once:</pre>				^
3				
<pre>void loop() {     // put your main code here, to run repeat</pre>	edly:			
}				
				~
1 M-Du	uino family, M-C	Duino 38A	R+ on CC	M27

T: +62 811540593 M: support@rdd-tech.com W: www.rdd-tech.com



#### **Pengenalan Arduino IDE**

#### 3. Pengenalan tools pada Arduino IDE

Saat Arduino IDE pertama kali dijalankan akan muncul jendela seperti gambar di *slide* selanjutnya:



## **Pengenalan Arduino IDE**

**RDD** Technologies



- Merupakan menu di Arduino IDE. Di bagian ini terdapat 1. Menu "*File*" yang digunakan untuk mengeksekusi file \*.ino. Menu "*Edit*" yang digunakan untuk mengedit program yang sedang ditulis di *sketch*. Menu "**Sketch**" yang salah satu fungsinya digunakan untuk mem-verify program dan meng-upload program ke board arduino. Menu "Tools" untuk memanggil alat-alat pendukung seperti Serial Monitor dan Serial Plotter. Dan yang terakhir vaitu menu "Help".
- Merupakan shortcut penting untuk memudahkan dalam 2. melakukan pemograman dengan Arduino IDE. Dimulai dari kiri shorcut tersebut adalah : "Verify", "Upload", "New Sketch", "Open Sketch", "Save Sketch" dan "Serial Monitor".
- 3. Merupakan nama *sketch* dari *file* yang dipilih
- 4. Merupakan *sketch editor*, pada bagian ini kita menulis program dengan bahasa C.
- 5. Merupakan jendela informasi saat kita melakukan sesuatu pada Arduino IDE.
- 6. Merupakan jendela untuk mengetahui adanya kesalahan pada program yang kita buat.

## D Modul Praktikum Arduino MKR1000

**RDD** Technologies

- Module praktikum dibuat untuk memudahkan penggunaan Arduino, mudah untuk menghubungkan Arduino dengan sensor-sensor atau dengan aktuator
- Tidak perlu melakukan soldering, hanya dengan menggunakan konektor sensor/aktuator bisa terhubung dengan Arduino
- Layar LCD sudah terintegrasi dengan module.





#### Peralatan yang diperlukan untuk Praktikum

- Arduino MKR1000 loT Training Module
- *Power Supply* / Adaptor 7-12V, umumnya 12VDC
- Kabel microUSB (sama dengan yang dipakai pada HP android)
- 37 in 1 Sensor Kit







#### **Arduino MKR1000 IoT Training Module**

**RDD** Technologies





## Latihan 1: Menghubungan Arduino ke PC

- Nyalakan *Development board* Arduino MKR, pastikan lampu hijau indikasi power pada board menyala.
- Buka program Arduino IDE pada PC.
- Hubungkan kabel micro-USB ke PC dan ke Arduino.
- Pastikan Arduino terdeteksi dengan baik oleh PC, antara lain dengan mengecek pada windows device manager dan pada Arduino IDE.



#### Latihan 1: Menghubungan Arduino ke PC

**RDD** Technologies

📇 Device Manager	_	×
File Action View Help		
🔶 🔿 🕅 🗐 🗐 🛛 🖬 💭 💺	<b>X</b> 🖲	
V 🛃 DESKTOP-ATI8KK2		^
> 4 Audio inputs and outputs		
> 💻 Computer		
> 👝 Disk drives		
> 🏣 Display adaptors		
> 🛺 Human Interface Devices		
> 📷 IDE ATA/ATAPI controllers		
> 🚡 Imaging devices		
> Keyboards		
> III Mice and other pointing device	es	
> Monitors		
> 💭 Network adapters		
> Portable Devices		
V 🗒 Ports (COM & LPT)		
Arduino MKR1000 (COM5)	1	
Communications Port (CC	M1)	
Printer Port (LPT1)		
> 🚔 Print queues		
> 🛱 Printers		
> Processors		
> F Software components		
Software devices		
Sound, video and game control	ollers	
> 🍙 Storage controllers		
> To System devices		
🔋 📋 Universal Serial Rus controllers		× .

🥯 sketch_aug13a	Arduino 1.8.13 (Windows Store 1.8.42.0)		_		×	
ile Edit Sketch 1	Fools Help					
	Auto Format	Ctrl+T			Ø	
	Archive Sketch				-	
sketch_aug13;	Fix Encoding & Reload					
void setup()	Manage Libraries	Ctrl+Shift+I			1	1
// put your	Serial Monitor	Ctrl+Shift+M				
}	Serial Plotter	Ctrl+Shift+L				_
<pre>void loop() {</pre>	WiFi101 / WiFiNINA Firmware Updater					
// put your	Board: "Generic ESP8266 Module"	>				
}	Builtin Led: "2"	>				nı
	Upload Speed: "115200"	>				
	CPU Frequency: "80 MHz"	>				no
	Crystal Frequency: "26 MHz"	>				
	Flash Size: "1MB (FS:64KB OTA:~470KB)"	>				т
	Flash Mode: "DOUT (compatible)"	>				
	Flash Frequency: "40MHz"	>				
	Reset Method: "dtr (aka nodemcu)"	>				
	Debug port: "Disabled"	>				
	Debug Level: "None"	>				
	IwIP Variant: "v2 Lower Memory"	>				
	VTables: "Flash"	>				
	Exceptions: "Legacy (new can return nullptr)"	>				
	Erase Flash: "Only Sketch"	>				
	Espressif FW: "nonos-sdk 2.2.1+100 (190703)"	>				
	SSL Support: "All SSL ciphers (most compatible)"	>				
library is alr	Port	>	Serial port	s		
lbrary is alr	Get Board Info		COM1			
	Programmer	2	COM5 (Ar	duino N	/KR100	0)
1 WiFi 3	Burn Bootloader		o ESP8266 Modu dard (~20)	le on CO	2M10 20	-

T: +62 811540593

M: support@rdd-tech.com

W: www.rdd-tech.com



## Latihan 2: Upload program ke Arduino

- Pada aplikasi Arduino IDE Terdapat banyak contoh program, bisa dibuka dari menu File → Examples.
- Untuk pengenalan kita buka contoh program Blink. File→Examples→1.
   Basics→Blink.
- Untuk *upload* program ke Arduino MKR pilih menu *Sketch*  $\rightarrow$  *Upload*.
- Pastikan tidak ada *error* dan program yang di-*upload*.



### Arduino MKR1000 Hardware

- Arduino MKR1000 memiliki I/O yang dapat digunakan untuk membaca *input* dari sensor atau mengirim perintah ke aktuator
- Pada Arduino I/O dapat dikelompokan menjadi:
- 1. Digital Input
- 2. Digital Output
- 3. Analog Input
- 4. Analog Output / Pulse Output
- 5. Communication I/O (UART, SPI, I2C)



## Arduino MKR1000 I/O

- 1. Digital Input : berfungsi untuk membaca masukan dari sensor yang memiliki 2 keadaan (*state*), *On* atau *Off / Logic* 0 dan *Logic* 1. Misalnya : Saklar/*switch*/tombol
- 2. Digital Output: berfungsi untuk mengirim perintah ke aktuator yang memiliki 2 keadaan (*state*), *On* atau *Off / Logic 0* dan *Logic* 1. Misalnya : Lampu, Relay, Motor/Fan, *Valve*, sirine/*alarm*.
- 3. Analog Input: berfungsi untuk membaca masukan dari sensor yang berupa sinyal analog. Misalnya, sinyal analog berupa tegangan yang nilainya bisa berubah-ubah.
- 4. Analog Output: berfungsi untuk mengeluarkan tegangan yang nilainya bisa berubahubah
- 5. Pulse Output : Berfungsi untuk mengeluarkan sinyal *On* atau *Off* yang frekuensinya bisa diatur.
- 6. Communication I/O: berfungsi untuk bertukar data antara Arduino dengan perangkat lain, misalnya ke PC, *memory*, LCD *display*, Bluetooth atau bahkan mikrokontroler lainnya.



## Spesifikasi I/O pada Arduino MKR1000

- 1. Digital Input : 0 VDC (Logic 0/"OFF") dan 3.3 VDC (Logic 1/"ON")
- 2. Digital Output: 0 VDC (Logic 0/"OFF") dan 3.3 VDC (Logic 1/"ON")
- 3. Analog Input : 0 3.3 VDC
- 4. Analog Output : 0 3.3 VDC
- 5. Pulse Output : 0 VDC (Logic 0/"OFF") dan 3.3 VDC (Logic 1/"ON")
- 6. Communication I/O: 0 VDC (Logic 0/"OFF") dan 3.3 VDC (Logic 1/"ON")

## Spesifikasi I/O pada Arduino MKR1000

**RDD** Technologies



				ARDUINO.CC
Ground	Internal Pin	💋 Digital Pin	Microcontroller's Port	
Power	SWD Pin	🗾 Analog Pin		BY SA
LED	Other Pin	Default		This work is licensed under the Creative Common Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit http://creativecommons org/licenses/lysa4/L or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA

A0 – A6 = Analog Input

- D0 D14 = Digital Input / Output
- Analog input bisa difungsikan sebagai Digital Input/Output, Namun Digital Input/Output tidak bisa difungsikan sebagai analog Input.



## **Sensor dan Actuator Training Module**

**RDD** Technologies

- TCS34725 RGB Color Sensor+B40BB2:B38
- Heart Rate Monitor Sensor ٠
- **I2C Digital Wattmeter** ٠
- Conductivity Sensor Switch ٠
- **Digital Shake Sensor**
- Analog Grayscale Sensor •
- Analog LM35 Temperature Sensor ٠
- Analog Ambient Light Sensor ٠
- **Digital Vibration Sensor** •
- **Digital Tilt Sensor** ٠
- **Digital Capacitive Touch Sensor** •
- **Digital Magnetic Sensor**
- Analog Sound Sensor

- Analog Carbon Monoxide Sensor (MQ7)
- Analog Voltage Divider
- Digital Piezo Disk Vibration Sensor
- Analog Rotation Potentiometer Sensor
- Joystick Module
- Analog Flame Sensor
- Triple Axis Accelero
- **Digital Infrared Motion Sensor** •
- URM09 Analog Ultrasonic Sensor
- Analog Soil Moisture Sensor
- meter MMA7361
- Steam Sensor
- Digital Push Button (White)

- **Digital Push Button (Red**
- Digital Push Button (Yellow) •
- **Digital White LED Light Module**
- Digital RED LED Light Module •
- **Digital Green LED Light Module** •
- Digital Blue LED Light Module
- **Digital RGB LED Module**
- **Digital 5A Relay Module**
- **Digital Speaker Module**
- Vibration Motor Module
- 130 DC Motor Module •
- I2C 16x2 Arduino LCD with RGB **Backlight Display**

# Spesifikasi Sensor – Sensor pada Module

- Tegangan catu daya (*power supply*) : 5 VDC
- Rentang tegangan *output* data digital atau sinyal analog: 0-5 VDC
- Arduino MKR bekerja di tegangan 3.3V, dengan demikian tidak bisa menerima data dari sensor secara langsung, karena akan menyebabkan kerusakan pada perangkat tersebut.
- Arduino MKR1000 IoT Training Module sudah dilengkapi dengan *logic* converter yang mengkonversi tegangan dari 0-5 VDC ke 0-3.3 VDC.
   Sehingga sensor dan aktuator bisa langsung terhubung ke Arduino MKR1000.



#### **Arduino MKR1000 IoT Training Module**

**RDD** Technologies




### a) Tujuan

- 1) Memberikan pemahaman praktikal mengenai *digital input* dan *output*.
- 2) Memberikan pengetahuan dasar *hardware* Arduino.
- 3) Memberikan pemahaman praktikal mengenai control statement IF.

#### b) Alat dan bahan

**RDD** Technologies

- 1) Arduino MKR1000 Development Board @ 1 unit
- 2) DFRobot Push Button module @ 1 unit
- 3) DFRobot LED module @ 1 unit
- 4) DFRobot Digital Speaker Module @ 1 unit
- 5) DFRobot Relay Module @ 1 unit
- 6) RDD-Tech MKR IoT Training Module @ 1 unit
- 7) Kipas 220VAC atau lampu atau beban yang lain @ 1 unit
- 8) Kabel dan konektor @ secukupnya
- 9) Kabel USB Arduino MKR1000 Development Board @ 1 unit 10)Arduino IDE
- 11)PC @ 1 unit
- 12)Power Supply 12V @ 1 unit

### c) Teori Singkat

Pada bagian teori singkat ini akan dibahas beberapa materi yang akan digunakan pada latihan ini diantaranya sebagai berikut:

#### 1) Push button

*Push button* merupakan salah satu saklar mekanik yang memiliki dua jenis pensaklaran yakni *push to make* dan *push to break*. *Push button* dengan jenis pensaklaran *push to make* akan menguhubungkan sirkuit ketika ditekan. *Push to break* melakukan kebalikannya yaitu akan memutuskan sirkuit jika ditekan.





#### 2) Light emitting diode (LED)

LED adalah sebuah dioda yang dapat memancarkan cahaya jika diberi tegangan melebihi tegangan bias majunya (*forward bias voltage*). Tegangan bias maju diberikan dengan cara memberi kaki anoda kutub positif *supply* (contoh: batre) dan kaki katoda kutub negatif *supply*. Besarnya tegangan yang dibutuhkan berbeda tiap warna pada LED.

**RDD** Technologies



Garis putus-putus pada gambar merupakan nilai *typical* untuk arus bias maju (*forward current*) LED sebesar 20mA.

Arus bias maju ini dapat dibatasi dengan memasang resistor secara seri dengan LED.

Pembatasan arus ini bertujuan supaya LED tidak cepat rusak dan menghemat konsumsi arus.



Berikut adalah cara menentukan besar nilai resistor yang dibutuhkan berdasarkan arus bias maju yang diinginkan.





#### 3) Speaker & Buzzer

Speaker merupakan salah satu komponen aktuator yang dapat menghasilkan suara dari sinyal-sinyal elektronika yang diberikan kepadanya. Suara ini dihasilkan dari membran yang tarik-ulur oleh kumparan dan magnet. Speaker terbagi beberapa jenis tergantung frekuensi dan aplikasinya yakni *bass, treble,* dan *tweeter*. Namun speaker tidak bisa berdiri sendiri atau membutuhkan komponen pendukung seperti *pre-amp, amplifier, audio tunner* dan lain sebagainya.



Berbeda halnya dengan *speaker, buzzer* merupakan aktuator yang dapat menghasilkan suara yang berasal dari kepingan piezoelektrik yang bergetar.

Elemen piezoelektrik akan bergetar jika diberi tegangan arus searah (DC), hal ini dikenal sebagai efek piezoelektrik.



#### 4) Relay

Relay merupakan salah satu saklar elektromekanik yang banyak dan paling umum digunakan di bidang industri. Relay digunakan dengan cara memberi tegangan eksitasi (*exciter*) pada kedua ujung kumparannya (*coil*). Setelah kumparan tereksitasi (*excited*) maka akan membuat sebuah medan magnet yang dapat menarik kontak sehingga kontak tersebut dapat menghubungkan sirkuit. Tegangan eksitasi ini dapat diberikan oleh perangkat-perangkat pengendali (*controller*) seperti PLC (*Programmable Logic Controler*), mikrokontroler, komparator analog dan lain sebagainya.



```
4) Control statement "IF"
```

Perintah "if" merupakan dasar dari semua bahasa pemograman. Perintah tersebut dapat melakukan sesuatu jika kondisi yang ada dalam kurung "()" terpenuhi. Perintah "if" pada arduino IDE dapat ditulis seperti dibawah ini.

```
if (kondisi)
{
//yang akan dilakukan jika kondisi terpenuhi
}
```

Statement "if" memiliki variasi yakni, if-else, if-else if-else.



If-else berfungsi sama dengan statement "if" tetapi jika kondisi tidak terpenuhi maka program juga akan melakukan sesuatu.

if (kondisi)
{
 //yang akan dilakukan jika kondisi terpenuhi
}
else
{
 //yang akan dilakukan jika kondisi tidak terpenuhi
}



If-else if-else sama seperti "if-else" namun pada *statement* ini jika kondisi pertama tidak terpenuhi maka kondisi ke dua akan dicek oleh program, jika kondisi kedua tidak terpenuhi maka kondisi ke tiga akan dicek, begitu seterusnya sampai dengan *statement* terakhir.

```
if (kondisi)
{
    //Akan dilakukan jika kondisi terpenuhi
}
else if (kondisi ke 2)
{
    //Akan dilakukan jika kondisi ke 2 terpenuhi
}
else
{
    //Akan dilakukan jika semua kondisi tidak terpenuhi
}
```

### d) Langkah-langkah keselamatan

- Pastikan pemasangan kabel penguhubung telah benar sesuai dengan kode warna yang diberikan modul DFRobot.
- 2) Pastikan polaritas *power supply* tidak terbalik.
- 3) Jika menggunakan beban dengan tegangan jala-jala 220VAC. Pastikan tidak ada kabel yang terkelupas atau serabut yang mencuat saat memasang beban pada modul relay.
- 4) Lepas catu dari papan mikrokontroler baik dari kabel USB maupun dari *power supply* sebelum mengubah rangkaian.

### e) Langkah-langkah latihan

- 1) Siapkan alat dan bahan seperti yang telah dituliskan diatas.
- 2) Siapkan dan hubungkan catu daya serta kabel USB pada Arduino MKR1000 IoT Training Module.
- 3) Rangkailah rangkaian seperti pada gambar latihan 3.1 di bawah ini.



- 4) Coba nyalakan LED dengan tombolnya masing-masing, contoh nyalakan LED1 dengan menekan PB1.
- 5) Modifikasi rangkaian gambar latihan 3.1 dengan menambahkan satu buah LED dan coba nyalakan LED tersebut dengan kombinasi dari tombol-tombol yang telah terpasang sebelumnya seperti pada gambar latihan 3.2 di bawah ini.





6) Rangkailah rangkaian seperti pada gambar latihan 3.3 di bawah ini.



7) Coba buat modul speaker mengeluarkan suara selama 1 detik dengan satu tombol. \*petunjuk: gunakan perintah tone(pinArduino, frekuensi, durasi) & noTone(pinArduino).



8) Rangkailah rangkaian seperti pada gambar latihan 3.4 di bawah ini.



9) Coba nyalakan kipas atau beban yang lain menggunakan modul relay yang diperintah oleh satu tombol.



10)Buatlah kombinasi dari semua rangkaian yang telah dicoba. Be creative!

### a) Tujuan

- 1) Memberikan pengetahuan praktikal mengenai *digital input* dengan hasil proses berupa *analog output*.
- 2) Memberikan pengetahuan dasar mengenai *hardware* Arduino.
- 3) Memberikan pengetahuan teori dan praktikal mengenai MOSFET (*Metal Oxide Semiconductor Field Effect Transistor*).
- 4) Memberikan pengetahuan praktikal pengendalian aktuator dengan PWM (*Pulse Width Modulation*).

### b) Alat dan bahan

- 1) Arduino MKR1000 Development Board @ 1 unit
- 2) DFRobot Digital Push Button Module @ 3 unit
- 3) DFRobot 130 DC Motor Module V1.0 @ 1 unit
- 4) RDD Tech Arduino MKR IoT Training Module @ 1 unit
- 5) Kabel dan konektor @ secukupnya
- 6) Kabel USB Arduino Development Board @ 1 unit
- 7) Arduino IDE
- 8) PC @ 1 unit
- 9) Power Supply 12V @ 1 unit

### c) Teori Singkat

Pada bagian teori singkat ini akan dibahas beberapa materi yang akan digunakan pada latihan ini diantaranya sebagai berikut:

#### 1) MOSFET (Metal Oxide Semiconductor Field Effect Transistor)

*Metal oxide semiconductor field effect tran*sistor (MOSFET) memiliki kaki atau terminal *source, gate,* dan *drain* seperti yang ditunjukan gambar di slide berikutnya.

MOSFET berbeda dengan JFET, perbedaannya terletak pada gate yang terinsulasi dari kanalnya (*channel*). Oleh karena itu MOSFET memiliki arus gate yang lebih kecil daripada JFET. MOSFET juga disebut IGFET atau *insulated field effect transistor*.

(A. Malvino dan D. J. Bates, Electronic Principles, New York: McGraw-Hill, 2006)

**RDD** Technologies



MOSFET terbagi menjadi dua jenis yakni, tipe depletion-mode dan tipe enhancement-mode seperti yang ditunjukkan pada gambar disamping. A dan B merupakan tipe enhancement-mode sedangkan C dan D merupakan depletion-mode.

MOSFET berjenis *enhancement-mode* sangat luas digunakan baik dalam sirkuit diskrit maupun sirkuit terintegrasi (IC). Pada sirkuit diskrit MOSFET jenis ini sering digunakan untuk menyalakan atau menghidupkan beban dengan kebutuhan arus yang besar, sedangkan pada sirkuit terintegrasi digunakan sebagai saklar digital yang merupakan proses dasar dari komputer-komputer modern.

MOSFET berjenis *depletion-mode* digunakan pada sirkuit komunikasi front-end berfrekuensi tinggi seperti penguat RF.

(Radio Frequency) (A. Malvino dan D. J. Bates, Electronic Principles, New York: McGraw-Hill, 2006)

Selain jenis modenya, MOSFET juga terbagi dua berdasarkan jenis kanalnya (*channel*) yakni MOSFET kanal P (*P-channel MOSFET*) dan MOSFET kanal N (*N-Channel MOSFET*).

Perbedaan yang paling sederhana dari keduanya adalah arah arus yang mengalir. Pada kanal N arus mengalir dari *drain* ke *source* dan arus mengalir sebaliknya pada kanal P yakni dari *source* ke *drain*.



MOSFET memiliki beberapa kelebihan diantaranya sebagai berikut:

MOSFET memiliki beberapa kelebihan diantaranya adalah *fast switching*, disipasi daya yang rendah, dan mampu beroperasi pada suhu kerja maksimum 175°C. (M. H. Rashid, Power Electronics Handbook, Canada: Academic Press, 2001)

Prinsip kerja MOSFET berjenis *N-channel enhancement-mode* adalah sebagai berikut:

- Apabila VDS < VGS-VTh untuk VGS > VTh maka MOSFET bekerja didaerah aktif namun tidak mengalirkan arus dari *drain* ke *source*.
- Apabila VDS > VGs-VTh untuk VGS > VTh maka MOSFET bekerja didaerah aktif mengalirkan arus dari *drain* ke *source*.
- Apabila VGS < VTh maka MOSFET tidak bekerja (non-aktif).
- Apabila VGS > VTh maka MOSFET bekerja didaerah aktif.



Dibawah ini merupakan kurva karakteristik N-channel enhancement-mode MOSFET:



#### 2) PWM (Pulse Width Modulation)

PWM merupakan sebuah pendekatan perubahan sinyal digital (pulse) menjadi sinyal analog dengan memanipulasi *duty cycle* (0 - 100%) sinyal digital tersebut. PWM banyak digunakan pada mikrokontroler termasuk mikrokontroler Arduino.

Duty cycle adalah perbandingan waktu "ON" dan "OFF" sinyal digital tersebut.

**Duty cycle** (%) =  $\frac{Ton}{Periode} \times 100\%$ 

Misalnya sebuah sinyal PWM dengan duty cycle 50% pada frekuensi 50Hz. Jika gelombang memiliki frekuensi 50Hz maka periode gelombang tersebut adalah 20ms. Dengan demikian gelombang tersebut memiliki Ton selama 10ms dan Toff selama 10ms.

PWM pada mikrokontroler memiliki resolusi tergantung pada jenis dan tipe mikrokontrolernya. Contohnya Arduino Uno, Leonardo, Mega dan MKR1000 memiliki PWM dengan resolusi 8 bit (0 sampai 255).



**RDD** Technologies



Time

**RDD** Technologies



PWM pada Arduino hanya bisa dikeluarkan pada pin I/O dengan tanda bendera (~).

Arduino MKR1000 semua pin PWMnya (pin D2, D3, D4, dan D5) memiliki frekuensi yang sama sekitar 733Hz.



Perintah untuk mengeluarkan PWM pada pin I/O pada Arduino adalah:

analogWrite(pin, value);

### d) Langkah-langkah keselamatan

- 1) Pastikan pemasangan kabel penguhubung telah benar sesuai dengan kode warna yang diberikan modul DFRobot.
- 2) Pastikan polaritas *power supply* tidak terbalik.
- 3) Jika menggunakan beban dengan tegangan jala-jala 220VAC. Pastikan tidak ada kabel yang terkelupas atau serabut yang mencuat saat memasang beban pada modul relay.
- 4) Lepas catu dari papan mikrokontroler baik dari kabel USB maupun dari *power supply* sebelum merubah rangkaian.

#### e) Langkah-langkah latihan

- 1) Siapkan alat dan bahan seperti yang telah dituliskan diatas.
- 2) Siapkan dan hubungkan catu daya serta kabel USB pada Arduino MKR1000 IoT Training Module.
- 3) Rangkailah rangkaian seperti pada gambar latihan 4.1.



4) Buatlah satu *button* untuk menyalakan, satu untuk memadamkan, dan satu lagi untuk menambah kecepatan putaran motor dan tampilkan datanya pada *serial monitor*.



### a) Tujuan

- 1) Memberikan pengetahuan praktikal mengenai *analog input* dengan hasil proses berupa *digital output*.
- 2) Memberikan pengetahuan dasar mengenai hardware Arduino.
- 3) Memberikan pengetahuan mengenai pengolahan dasar sinyal analog.

## Latihan 5 : Analog Input dan Digital Output

b) Alat dan bahan

**RDD** Technologies

- 1) Arduino MKR1000 Development Board @ 1 unit
- 2) DFRobot Analog Rotation Potentiometer Sensor @ 1 unit
- 3) DFRobot LED module @ 1 unit
- 4) DFRobot Digital Speaker Module @ 1 unit
- 5) DFRobot Relay Module @ 1 unit
- 6) RDD Tech Arduino MKR IoT Training Module @ 1 unit
- 7) Kipas 220VAC atau lampu atau beban yang lain @ 1 unit
- 8) Kabel dan konektor @ secukupnya
- 9) Kabel USB Arduino Development Board @ 1 unit
- 10)Arduino IDE
- 11)PC @ 1 unit
- 12)Power Supply 12V @ 1 unit

# Latihan 5 : Analog Input dan Digital Output

### c) Teori Singkat

Pada bagian teori singkat ini akan dibahas beberapa hal yang akan digunakan pada latihan ini diantaranya sebagai berikut:

1) Potensiometer

Potensiometer merupakan resistor tiga kaki yang memiliki resistasi yang dapat diubah-ubah dengan cara memutar knobnya. Potensiometer sejatinya merupakan dua resistor yang dihubungkan, berikut merupakan simbolnya.



Nilai resistansi dari titik A ke C adalah konstan, yakni sesuai dengan besar resistansi total potensiometer, contohnya pada modul analog rotation potensiometer DFRobot memiliki resistansi sebesar 10K ohm. Resistansi titik A ke B dan B ke C dapat berubahubah dengan cara memutar knob namun total resistansi dari keduanya tidak akan melebihi resistansi total (Rt = Rab + Rbc). Titik B ini merupakan komponen wiper yang tertaut pada knob. Titik A ke C merupakan gulungan kawat tembaga atau bentangan film karbon. Dengan memutar knob berarti menggeser posisi wiper pada diantara panjang gulungan kawat, hal ini yang mempengaruhi nilai resistansi titik AB dan BC.







#### 2) Pengolahan sinyal analog

Fenomena fisika alam seperti temperatur, intensitas cahaya, tekanan udara, aliran debit air di sungai, dan lain sebagainya dapat disampling menjadi sinyal-sinyal analog yang bisa dibaca oleh mikrokontroler melalui analog input. Sinyal-sinyal analog ini bersifat kontinu, oleh karena itu mikrokontroler perlu mendiskritkan sinyal-sinyal analog agar dapat diolah olehnya. Komponen pendiskrit sinyal ini (ADC atau *Analog to Digital Converter*) biasanya telah terpasang didalam mikrokontroler. Biasanya resolusi ADC dinyatakan dengan bit.

Contohnya 10bit ADC memiliki rentang nilai dari 0 sampai 1023. Jika melakukan pembacaan sinyal dengan nilai maksimal sebesar 5V dan nilai minimal 0V, maka jika diberikan nilai 2.5V mikrokontroler akan membaca analog input sebesar 511. Semakin besar resolusi ADC maka sinyal yang terbaca oleh mikrokontroler akan semakin mulus (*smooth*) dan semakin mendekati nilai sesungguhnya (presisi).





Dikarenakan mikrokontroler merupakan sebuah sistem digital dan hanya mengerti sinyal digital (sinyal digital merupakan sinyal diskrit), maka sinyal analog harus diubah menjadi sinyal diskrit terlebih dahulu sebelum dapat diolah oleh mikrokontroler dengan cara melakukan *signal sampling*.


Untuk membaca sinyal analog melalui pin analog, pin tidak perlu dikonfigurasi seperti pada pin digital.

Untuk resolusi, pada board tertentu contohnya MKR1000 memiliki resolusi ADC 12 bit, untuk menggunakan resolusi ADC 12 bit (0-4095) tulislah perintah berikut pada void(setup):

# analogReadResolution(12); analogReference(AR\_DEFAULT);

Jika tidak ditulis, maka pembacaan akan dilakukan dengan mode default yaitu resolusi ADC 10 bit (0-1023).

Perintah untuk melakukan pembacaan analog input pada Arduino adalah:

analogRead(pin);

# Latihan 5 : Analog Input dan Digital Output

- d) Langkah-langkah keselamatan
  - 1) Pastikan pemasangan kabel penguhubung telah benar sesuai dengan kode warna yang diberikan modul DFRobot.
  - 2) Pastikan polaritas *power supply* tidak terbalik.
  - 3) Jika menggunakan beban dengan tegangan jala-jala 220VAC. Pastikan tidak ada kabel yang terkelupas atau serabut yang mencuat saat memasang beban pada modul relay.
  - 4) Lepas catu dari papan mikrokontroler baik dari kabel USB maupun dari *power supply* sebelum merubah rangkaian.



### e) Langkah-langkah latihan

- 1) Siapkan alat dan bahan seperti yang telah dituliskan diatas.
- 2) Siapkan dan hubungkan catu daya serta kabel USB pada Arduino MKR1000 IoT Training Module.
- 3) Rangkailah rangkaian seperti pada gambar latihan 5.1 dibawah ini.



4) Coba lakukan pembacaan nilai pada potensiometer (kabel biru) dengan menggunakan perintah analogRead, ubah nilai tersebut menjadi satuan (tegangan) lalu tampilkan datanya pada serial monitor.



5) Tambahkan LED atau buzzer atau relay pada rangkaian seperti pada gambar latihan 5.2 dibawah, nyalakan dengan menggunakan *input* dari potensiometer, lalu tampilkan datanya pada serial monitor.





6) Anggaplah potensiometer ini sebagai sensor panas atau sensor-sensor analog lainnya dan buatlah sebuah sistem alarm dengan menggunakan led sebagai komponen peringatan level aman, *buzzer* sebagai level siaga dan relay sebagai komponen yang dapat menyalakan atau memadamkan aktuator jika keadaan naik menjadi genting dan tampilkan pula datanya pada serial monitor. Rangkaian dapat dilihat pada gambar 5.3 dibawah ini.



# Latihan 6 : Analog Input dan Analog Output

## a) Tujuan

- 1) Memberikan pengetahuan praktikal mengenai *analog input* dengan hasil proses berupa *analog output*.
- 2) Memberikan pengetahuan dasar mengenai hardware Arduino.
- 3) Memberikan pengetahuan dasar mengenai fungsi map.



## b) Alat dan bahan

- 1) Arduino MKR1000 Development Board @ 1 unit
- 2) DFRobot Analog Rotation Potentiometer Sensor @ 1 unit
- 3) DFRobot 130 DC Motor Module V1.0 @ 1 unit
- 4) RDD Tech Arduino MKR IoT Training Module @ 1 unit
- 5) Kabel dan konektor @ secukupnya
- 6) Kabel USB Arduino Development Board @ 1 unit
- 7) Arduino IDE
- 8) PC @ 1 unit
- 9) Power Supply 12V @ 1 unit



#### c) Teori Singkat

Pada bagian teori singkat ini akan dibahas beberapa hal yang akan digunakan pada latihan ini diantaranya sebagai berikut:

1) Map Function

Arduino IDE memiliki fungsi yang dapat me-mapping dua nilai yang memiliki rentang yang berbeda misalnya nilai a memiliki nilai paling rendah 0 dan paling tinggi 1023 dan nilai b memiliki nilai paling rendah 0 dan paling tinggi 255. Namun *map function* ini akan berfungsi dengan baik jika kedua nilai bersifat linear.

Untuk memanggil fungsi ini kita dapat menuliskan seperti dibawah ini:

map(nama variabel, nilai min a, nilai max a, nilai min b, nilai max b);

# Latihan 6 : Analog Input dan Analog Output

- d) Langkah-langkah keselamatan
  - 1) Pastikan pemasangan kabel penguhubung telah benar sesuai dengan kode warna yang diberikan modul DFRobot.
  - 2) Pastikan polaritas *power supply* tidak terbalik.
  - 3) Jika menggunakan beban dengan tegangan jala-jala 220VAC. Pastikan tidak ada kabel yang terkelupas atau serabut yang mencuat saat memasang beban pada modul relay.
  - 4) Lepas catu dari papan mikrokontroler baik dari kabel USB maupun dari *power supply* sebelum merubah rangkaian.

# Latihan 6 : Analog Input dan Analog Output

### e) Langkah-langkah latihan

- 1) Siapkan alat dan bahan seperti yang telah dituliskan diatas.
- 2) Siapkan dan hubungkan catu daya serta kabel USB pada Arduino MKR1000 IoT Training Module.
- 3) Rangkailah rangkaian seperti pada gambar latihan 6.1 dibawah ini.



4) Coba lakukan pembacaan nilai pada potensiometer (kabel biru) dengan menggunakan perintah analogRead, ubah nilai tersebut menjadi satuan (tegangan) dengan menggunakan fungsi map dan tampilkan datanya pada serial monitor.



 Tambahkan modul DC motor seperti pada gambar latihan 6.2 dibawah ini. Lakukan konversi nilai ADC menjadi nilai PWM menggunakan fungsi map dan tampilkan datanya pada serial monitor.





## Latihan 5 : Analog Input

T: +62 811540593 M: support@rdd-tech.com W: www.rdd-tech.com



## Pengenalan Arduino IDE

4. Materi singkat mengenai struktur dan alur eksekusi program pada Arduino IDE

#### a) Struktur

Struktur dasar dari bahasa pemrograman arduino hanya terdiri dari 2 bagian yaitu setup() dan loop().

```
void setup()
{
   // Statement;
}
void loop()
{
   // Statement;
}
```



Bagian setup() merupakan bagian yang hanya dijalankan satu kali saja yaitu pada saat program pertama dijalankan. Bagian ini berfungsi sebagai pedefinisian suatu perangkat, seperti pedefinisian pin,komunikasi,SPI dan lain-lain. Contoh:

```
void setup()
{
    pinMode(13,OUTPUT); // mengset pin 13 di board arduino sebagai output
}
```

Bagian loop() merupakan bagian yang akan terus dijalankan dan akan melakukan instruksiinstruksi/program yang ada di dalam loop() secara berurutan. Contoh:

```
void loop()
{
    digitalWrite(13, HIGH); // nyalakan pin 13
    delay(1000); // delay selama 1000 ms
    digitalWrite(13, LOW); // matikan pin 13
    delay(1000); /// delay selama 1000 ms
}
```



5. Mencoba dan mengunggah (*upload*) program sederhana

Kali ini kita akan mencoba mengunggah program ke dalam *Board* Arduino. Langkah-langkah yang harus dilakukan adalah sebagai berikut:

- a) Buka aplikasi Arduino IDE.
- b) Tekan tombol *File -->Example-->*01.*Basic-->Blink*, jika berhasil maka akan muncul jendela seperti gambar di *slide* selanjutnya.





T: +62 811540593 M: support@rdd-tech.com

W: www.rdd-tech.com



c) Sebelum mengunggah Blink.ino ke *Board* Arduino, kita harus memilih terlebih dahulu Board Arduino yang kita pakai. yaitu dengan cara tekan menu *Tools--*>*Board*, lalu pilih Board Arduino yang kita pakai, kita memakai Arduino MKR1000, maka pilih Board Arduino MKR1000.



•						
🥺 Blink   Arduino	1.8.10	- 0		×		
File Edit Sketch To	ools Help					
	Auto Format	Ctrl+T		0	-	
	Archive Sketch			~		
Blink	Fix Encoding & Peload			-		
/*	The Encoding of Reload	01.0170.1		~		
Blink	Manage Libraries	Ctrl+Shift+I				
	Serial Monitor	Ctrl+Shift+M				
Turns an LE	Serial Plotter	Ctrl+Shift+L	epe	ate		
Most Arduin WiFi101 / WiFiNINA Firmware Updater				Board	s Manager	
it is attac	· · ·		Δ.			
the correct	Board: "Arduino MKR1000"	;		Ardui	no SAMD (32-bits ARM Cortex-M0+) Board	
If you want	Port			Andui		
https://www	S://www Get Board Info			Ardui		
	Programmer "AVRISP mkll"	,		Ardui	no Zero (Native USB Port)	
modified 8	Programmer. Avrise mixin			Ardui	no MKR1000	
by Scott Fi Burn Bootloader				Ardui	no MKRZERO	
by Arturo Guadalupi				Arduino MKR WiFi 1010		
modified 8 Sep 2016				Arduino NANO 33 IoT		
by Colby Newm	lan			Ardui	no MKR FOX 1200	
This example	code is in the public domain			Ardui	no MKR WAN 1300	
				Ardui	no MKR WAN 1310	
د 🛛				Ardui	no MKR GSM 1400	
				Ardui	no MKR NB 1500	
				Ardui	no MKR Vidor 4000	
				Adafr	uit Circuit Playground Express	
				Ardui	no MO Pro (Programming Port)	
				Andui	no MO Pro (Programming Port)	
		Arduino MKR1000		Ardui	no MO Pro (Native OSB Port)	
				Ardui	no M0	
				Ardui	no Tian	
				Ardui	no AVR Boards	
				Ardui	no Yún	
				Ardui	no/Genuino Uno	
				Ardui	no Duemilanove or Diecimila	
				Ardui	no Nano	

T: +62 811540593 M: support@rdd-tech.com W: www.rdd-tech.com



- d) Setelah Blink.ino terbuka, kita akan mengunggah program tersebut kedalam Board Arduino yaitu dengan cara tekan menu *sketch*, lalu tekan tombol *upload*, intruksi *upload* pada arduino IDE yaitu akan melakukan proses *verify program* terlebih dahulu, lalu menggunggahnya ke Board Arduino jika program tidak memiliki *error*.
- e) Jika proses *upload* telah selesai maka akan muncul tulisan "*Done Uploading*" pada bagian jendela informasi dan LED pada Board Arduino berkedip.



# Pengenalan Internet of Things (IoT)

### 1. Sejarah Singkat Internet of Things

Pada awal tahun 1990-an, internet ditemukan dan dikenal sebagai "Internet of Computers" sebagai jéjaring komunikasi dunia dengan layanan "world wide web". Tahun-tahun selanjutnya, akibat dari pengembangan Web 2.0, istilah "Internet of Computers" berganti menjadi "Internet of People" karena terdapat miliaran manusia yang saling terkoneksi melalui situs sosial atau media sosial. Definisi Internet semakin meluas tiap harinya akibat dari pengintegrasian Micro Electro Mechanical devices (MEMs), teknologi komunikasi nirkabel dan komputer-komputer personal yang semakin mengecil dengan kemampuan memroses dan menyimpan data lebih besar dari sebelumnya. Pada tahun 2005, dalam laporan International Telecommunication Union (ITÚ) menyatakan bahwa "Internet of Things will connect the world's objects in sensory and intelligent manner". (Internet of Things Services, Applications, Issues, and Challenge - IGI Global -2017 - Hlm 353)



### 2. Definisi Internet of Things

Seperti yang telah dijelaskan oleh kalimat terakhir pada kutipan diatas, "world's objects" ini merupakan kata "Things" dalam "Internet of Things". Things ini bisa berupa apa saja yang terkoneksi secara "sensory" (bisa melakukan pengindraan) dan "intelligent" (memikirkan / memroses data). Namun intelligent disini lebih menjurus pada kecerdasan buatan (Artificial Intelligence). Secara harfiah, Internet of Things berarti jejaring dari banyak benda yang saling terkoneksi dalam jejaring internet yang dapat melakukan pengindraan (sensory) dan memroses data baik secara otomatis dengan menggunakan kecerdasan buatan atau tidak (intelligence).



Adapula definisi Internet of Things yang lainnya adalah sebagai berikut:

Internet of Things adalah jaringan dari objek-objek fisik seperti pirantipiranti yang dapat dipakai, peralatan rumah, sistem keamanan, mesinmesin manufaktur dan lain sebagainya yang tergabung (*embedded*) dengan komponen "pintar" seperti mikroprosesor, penyimpanan data, perangkat lunak, sensor, aktuator dan lain-lain yang terhubung juga dengan sistem atau perangkat lain melalui internet.

(IoT Solution For Dummies - Lawrence C Miller - 2018 - Hlm 3)



### 3. Definisi Things

Berdasarkan dari kedua kutipan diatas, *Internet of Things* adalah sebuah jaringan yang terdiri dari "*Things*" yang terhubung satu sama lain melalui internet. Setiap "*Things*" tergabung atau terpasang bersama komponen "pintar" yang tak hanya mampu melakukan pengindraan dan memroses data namun juga melakukan tindakan dari hasil data yang diproses.



Things bisa berupa banyak hal, handphone yang terkoneksi dengan internet sudah dapat dikatakan "things".

Sebagai contoh, *handphone* sekarang telah dilengkapi GPS yang terintegrasi dengan internet map. Handphone mengirimkan koordinatnya sesuai pembacaan GPS ke internet map (contohnya: googlemap).

Data GPS diolah oleh server googlemap lalu dikirimkan kembali untuk ditampilkan pada aplikasi googlemap yang terpasang pada *handphone*. Sehingga memudahkan penggunanya dalam berkendara dan berpergian.





### 4. Arsitektur Internet of Things

Pada umumnya arsitektur internet of things terdiri dari 3 bagian yaitu hardware, middleware, dan presentation. (Internet of Things Services, Applications, Issues, and Challenge - IGI Global - 2017 - Hlm 355)

Hardware bisa berupa sensor node, smart appliance, processor dan lain sebagainya. Middleware merupakan perangkat lunak yang berfungsi memperantarai hardware dengan internet. Middleware juga sering disebut sebagai "software glue" karena dapat menyatukan dua sistem operasi yang berbeda agar dapat berkomunikasi satu sama lain. Presentation adalah perangkat yang menampilkan data secara visual bisa berupa web, layar lcd, dan lain sebagainya.



## 5. Aplikasi Internet of Things dalam dunia industri

Internet of Things dapat diaplikasikan sebagai sistem monitoring dan kendali mesin-mesin industri. Pada umumnya sistem ini terdiri dari beberapa bagian yakni sensor, mikroprosesor atau mikrokontroler, aktuator, gateway, database, dan display. Contohnya seperti pada slide selanjutnya:



## 5. Aplikasi Internet of Things dalam dunia industri

Internet of Things dapat diaplikasikan sebagai sistem monitoring dan kendali mesin-mesin industri. Pada umumnya sistem ini terdiri dari beberapa bagian yakni sensor, mikroprosesor atau mikrokontroler, aktuator, gateway, database, dan display. Contohnya seperti pada slide selanjutnya:





User dapat memonitor dan mengendalikan plant (mesin-mesin industri) melalui internet dengan mengunjungi *website* dengan *browser smartphone* atau PC.

Widget-widget dalam website mengambil dan menyimpan perintah atau sinyal kendali yang mengendalikan aktuator serta mengambil hasil pengukuran besaran fisika oleh sensor yang tersimpan di dalam cloud database.

Gateway disini bisa berupa modul-modul WiFi atau Ethernet yang dapat diintegrasikan dengan mikroprosesor (uP) atau mikrokontroler (uC) yang terpasang bersama sensor dan aktuator.

Sedangkan uP atau uC berfungsi sebagai pengolah data sensor dan menyimpannya ke dalam *cloud database* serta menjalankan perintah yang diberikan lewat *website* oleh *user* yang telah tersimpan di dalam *cloud database*.



## Dasar Pemrograman menggunakan Arduino IDE

## 1. Data Type

Tipe data pada arduino memiliki kesamaan dengan tipe data yang dimiliki oleh bahasa c/c++, karena pada dasarnya bahasa pemograman arduino merupakan bahasa c /c++ yang telah dimodifikasi. Setiap *Board* Arduino memiliki memory yang kecil, maka dari itu pemilihan tipe data yang tepat akan membuat pemakaian *memory* lebih efisien. Berikut beberapa tipe data yang ada di Arduino IDE:



Tipe Data	Ukuran Bit	Rentang Nilai
bool	8	True atau False
char	8	-128 sampai 127
unsigned char	8	0 sampai 255
Byte	16	0 sampai 255
int	16	-32768 sampai 32767
unsigned int	16	0 sampai 65535
word	16	0 sampai 65535
long	32	-2,147,483,648 sampai 2,147,483,647
float	32	0 sampai 4,294,967,295
unsigned float	32	-3,4028235E38 sampai 3,4028235E38

Tipe data float dikhususkan untuk data yang memiliki bilangan dibelakang koma, contoh: 2,5; 0,55.



## 2. Operator

Operator dalam pemograman merupakan simbol yang menjelaskan kepada compiler untuk melakukan fungsi matematika dan logika khusus. Pada arduino terdapat banyak operator bawaan antara lain:

a) Operator Aritmatika

Operator aritmatika pada arduino meliputi:

- Operator Penugasan (=) berfungsi untuk menyimpan nilai yang ada di sebelah tanda (=) kanan ke dalam variabel yang ada di sebelah kiri tanda (=)
- Penjumlahan (+) berfungsi menjumlahkan operan.
- Pengurangan (-) berfungsi mengurangi operan .
- Perkalian (\*) berfungsi mengalikan operan.
- Pembagian (:) berfungsi membagi operan.
- Modulo (%) berfungsi menghasilkan sisa dari nilai bagi.



```
Contoh program:
int a;
int b;
int c;
void setup()
{
  a=10;
  b=5;
}
void loop ()
{
 c = a + b;
 c = a - b;
 c = a * b;
 c = a / b;
 c = a % b;
}
```



b) Operator Pembanding

Operator pembanding pada arduino meliputi:

- Sama dengan (==) berfungsi untuk membandingkan 2 nilai operan, jika kondisi 2 operan tersebut sama maka kondisi akan bernilai benar.
- Tidak Sama Dengan (!=) berfungsi untuk membandingkan 2 nilai operan, jika kondisi 2 operan tersebut tidak sama maka kondisi akan bernilai benar.
- Lebih kecil dari(<) berfungsi untuk membandingkan 2 nilai operan, jika nilai operan yang ada di sebelah kiri tanda (<) lebih kecil dari nilai operan yang ada di sebelah kanan tanda (<) maka kondisi akan bernilai benar.
- Lebih besar dari(>) berfungsi untuk membandingkan 2 nilai operan, jika nilai operan yang ada di sebelah kiri tanda (>) lebih besar dari nilai operan yang ada di sebelah kanan tanda (>) maka kondisi akan bernilai benar.
- Lebih kecil sama dengan dari(<=) berfungsi untuk membandingkan 2 nilai operan, jika nilai operan yang ada di sebelah kiri tanda (<=) lebih kecil atau sama dengan dari nilai operan yang ada di sebelah kanan tanda (<=) maka kondisi akan bernilai benar.
- Lebih besar sama dengan dari(>=) berfungsi untuk membandingkan 2 nilai operan, jika nilai operan yang ada di sebelah kiri tanda (=>) lebih besar atau sama dengan dari nilai operan yang ada di sebelah kanan tanda (=>) maka kondisi akan bernilai benar.



	if(a > b)
Contoh program:	{c = true;}
int a;	alaa
int b;	etse
bool c;	{c = false;}
void setup()	
{	if(a <= h)
a=10;	n (u <- b)
b=5;	{c = true;}
c=false;	else
}	{c = false;}
void loop ()	
{	
if(a == b)	if(a >= b)
{c = true;}	{c = true;}
else	else
{c = false;}	
if(a != b)	{c = false;}
{c = true;}	}
else	
{c = false;}	
if(a < b)	
{c = true;}	
else	
{c = false;}	



## c) Operator Boolean

Operator boolean merupakan operator yang mirip seperti gerbang logika/ Operator ini memiliki operasi AND,OR dan NOT. Berikut penjelasannya:

- AND (&&), merupakan operasi logika AND yang jika nilai kedua operan tidak bernilai nol maka kondisi menjadi benar.
- OR (||), merupakan operasi logika OR yang jika nilai salah satu dari 2 operan bernilai tidak nol maka kondisi menjadi benar
- NOT (!), merupakan operasi logika NOT yang digunakan untuk membalikan keadaan logika dari operan, jika nilai operan salah maka operator logika NOT akan membuat kondisi tersebut menjadi benar.



```
Contoh program:
int a;
int b;
bool c;
void setup()
{
  a=10;
  b=5;
  c=false;
}
void loop ()
if((a > b)&& (b < a))
   \{c = true;\}
  else
   {c = false;}
```


## 3. Control Statement

*Control Statement* merupakan bagian yang penting pada sebuah program. Sederhananya control statement berfungsi membuat program mengetahui apa yang harus dikerjakan jika terdapat *statement*/kondisi yang telah terpenuhi. *Control statement* pada arduino IDE yaitu "if" dan "switch".

a) If

Perintah "if" merupakan dasar dari semua bahasa pemograman. Perintah tersebut dapat melakukan sesuatu jika kondisi yang ada dalam kurung "()" terpenuhi. Perintah "if" pada arduino IDE dapat ditulis seperti pada slide dibawah ini.



```
if (kondisi)
{
    //yang akan dilakukan jika kondisi terpenuhi
}
```

Statement "if" memiliki variasi yakni, if-else, if-else if-else.



• If-else berfungsi sama dengan statement "if" tetapi jika kondisi tidak terpenuhi maka program juga akan melakukan sesuatu

```
if (kondisi)
{
    //yang akan dilakukan jika kondisi terpenuhi
}
else
{
    //yang akan dilakukan jika kondisi tidak terpenuhi
}
```



 If-else if-else sama seperti if namun pada statement ini jika kondisi pertama tidak terpenuhi maka kondisi ke dua akan di cek oleh program, jika kondisi kedua tidak terpenuhi maka kondisi ke tiga akan dicek, begitu seterusnya sampai dengan statement terakhir.



```
b) Switch
```

{

}

Perintah "switch" memilik cara kerja yang sama dengan perintah if...else if, perintah "switch" akan membandingkan nilai yang terdapat dalam "case", jika nilai yang di dalam "case" tersebut terpenuhi maka akan menjalankan perintah tertentu, setiap "case" akan di akhiri dengan "break". Contoh dari perintah "switch" adalah sebagai berikut.

```
switch (var) //variabel yang akan di bandingkan
```

```
case '2':
// jika var=2, perintah akan dijalankan
break;
```



# c) While

Perintah while adalah perintah pada arduino yang akan melakukan pengulangan program yang ada di dalam kurung kurawal "{}" sampai kondisi sudah tidak terpenuhi lagi. Bentuk umum dari perintah while yaitu :

```
while(var < 50)//kondisi yang harus terpenuhi agar terus berulang
{
     // program akan melakukan looping sebanyak 50 kali
     myVar=myVar + 1;
}</pre>
```



#### d) For loop

Perintah "for" digunakan untuk mengulang suatu blok program yang terdapat di dalam kurung kurawal ({}) setelah perintah "for". perintah "for" biasanya digunakan untuk menyederhanakan suatu perintah. Format dan contoh penulisan perintah "for" adalah sebagai berikut :

#### /\*

Pada contoh di atas dapat dijelaskan bahwa program tersebut akan mencetak nilai "b" di serial monitor sebanyak 10 kali. Hal itu terjadi karena nilai a setelah 10 kali pengulangan akan bernilai 10 yang artinya nilai a sudah tidak termasuk lagi dalam kodisi test "a<10", "a++" pada perintah tersebut maksudnya adalah "a=a+1". Jadi jika kita ingin menambah nilai variabel dengan 1 dapat memakai code "a++".



# 4. Delay

Delay() merupakan suatu fungsi yang terdapat dalam Arduino IDE yang berguna untuk menahan suatu program agar tidak melanjutkan perintah ke baris selanjutnya sampai delay yang telah ditentukan. Sintaks dari delay() yaitu:

delay( var);

Variabel "var" dalam kurung tersebut mempresentasikan delay yang akan di jalankan oleh program. Satuan untuk fungsi delay() adalah ms atau mili seconds. Dalam Arduino IDE tedapat juga delay yang satuannya adalah microseconds yaitu :

delayMicroseconds(var);



# 5. Millis()

Millis merupakan sebuah fungsi pada arduino IDE yang berfungsi sebagai pewaktu internal setiap 1 milisecond pada arduino. fungsi ini akan terus di jalankan walaupun arduino sedang mengerjakan perintah yang lain. Jadi dengan fungsi ini arduino dapat melakukan *multi-tasking*. Contoh pekerjaan yang itu ketika LED pertama berkedip selama 1000ms dan LED *multi-tasking* kedua berkedip selama 2000ms. Jika kita menggunakan fungsi delay, pekerjaan tersebut tidak akan berjalan dengan semestinya. Jika memakai millis, pekerjaan tersebut dapat dikerjakan dengan baik.

Contoh program blink dengan menggunakan fungsi millis():



T: +62 811540593 M: support@rdd-tech.com W: www.rdd-tech.com



```
unsigned long currentMillis;
unsigned long previousMillis;
unsigned int interval;
int ledstate;
void setup()
 Serial.begin(9600);
 interval=1000; //waktu yang ingin dicapai,satuannya ms
 pinMode(LED_BUILTIN,OUTPUT);
 ledstate=HIGH;
 digitalWrite(LED_BUILTIN,ledstate);
void loop()
currentMillis=millis();// menyimpan waktu saat ini pada ke variable "currentmillis"
 if (currentMillis-previousMillis>=interval)
previousMillis=currentMillis; // jika waktu interval telah
terpenuhi, nilai dari variabel "currentMillis" harus di simpan ke
dalah variabel "previousMillis"
```

```
Serial.println(previousMillis);
```

```
// mengecek kondisi LED, jika HIGH maka buat jadi LOW dan sebaliknya
```

```
if (ledstate==LOW)
```

{

```
ledstate=HIGH;
```

}

else

```
{
```

ledstate=LOW;

```
digitalWrite(LED_BUILTIN,ledstate);
```

# Pengenalan Input dan Output (I/O) Arduino

*Input* dan *output* pada mikrokontroler dibagi menjadi dua yaitu digital dan analog. Digital merupakan *input* atau *output* yang berisi isyarat biner (1 atau 0; High atau Low). Sedangkan analog berisi sinyal-sinyal analog, contohnya sinus, PWM, dan lain sebagainya.



## 1. Digital Input

Seperti yang telah dijelaskan di atas, *digital input* hanya menerima isyarat biner yakni 1 atau 0; High atau Low. Isyarat biner 1 berarti tegangan *supply* mikrokontroler (VCC/VDD) diberikan pada *digital input* mikrokontroler, tegangan *supply* ini bisa bervariasi tergantung jenis dan tipe mikrokontrolernya. Isyarat biner 0 berarti tegangan 0 V atau *ground* diberikan pada *digital input* mikrokontroler.

Telah disinggung sebelumnya, bahwa tegangan isyarat biner 1 adalah memberikan tegangan *supply* mikrokontroler pada *digital input* dan isyarat biner 0 adalah memberikan *ground* pada *digital* input. Namun kita juga dapat memberikan *logic* (isyarat biner) dengan cara memberikan tegangan yang spesifik (*voltage logic level*) pada digital input mikrokontroler, hal ini tergantung pada jenis mikrokontrolernya (CMOS atau TTL).



## Berikut merupakan voltage logic level untuk CMOS dan TTL :









Tiap-tiap jenis dan tipe mikrokontroler memiliki jumlah pin *digital input* yang berbeda-beda. Contohnya seperti mikrokontroler Arduino Uno dan Leonardo memiliki 14 pin yang dapat dikonfigurasi menjadi *digital input* atau *digital output*. Sedangkan Arduino Mega memiliki 54 pin yang dapat dikonfigurasi menjadi digital input atau digital output.

Sebelum melakukan pembacaan digital, pin input harus dikonfigurasi *mode input* pada bagian void(setup) dengan mengetik perintah berikut:

#### pinMode(pin, INPUT);

Perintah untuk melakukan pembacaan digital input pada Arduino adalah:

digitalRead(pin);



Berikut merupakan percobaan pembacaan *digital input* menggunakan mikrokontroler Arduino MKR1000 :





```
int buttonState = 0;
int buttonPin = 7;
void setup()
pinMode (buttonPin, INPUT);
Serial.begin(115200);
void loop()
buttonState = digitalRead(buttonPin);
```

```
if (buttonState == 1)
Serial.print("Button State = HIGH");
else
Serial.print("Button State = LOW");
delay(1);
```



© COM15 ─		×	💿 сом15 —		×
1	Se	end	1	S	end
Button State: LOW		^	Button State: HIGH		^
Button State: LOW			Button State: HIGH		
Button State: LOW			Button State: HIGH		
Button State: LOW			Button State: HIGH		
Button State: LOW			Button State: HIGH		
Button State: LOW			Button State: HIGH		
Button State: LOW			Button State: HIGH		
Button State: LOW			Button State: HIGH		
Button State: LOW			Button State: HIGH		
Button State: LOW			Button State: HIGH		
Button State: LOW			Button State: HIGH		
Button State: LOW			Button State: HIGH		
Button State: LOW			Button State: HIGH		
		$\checkmark$			$\sim$
Autoscroll Show timestamp Both NL & CR V 115200 baud V	Clear	output	Autoscroll Show timestamp Both NL & CR 🗸 115200 baud 🗸	Clear	output



Berikut merupakan percobaan pembacaan *analog input* menggunakan mikrokontroler Arduino MKR1000:





int potPin = A0; int potValue; void setup() analogReadResolution(12); analogReference(AR DEFAULT); Serial.begin(115200);

void loop()
{
 potValue = analogRead(potPin);
 Serial.print(potValue);
 delay(100);
}







## 3. Digital Output

Sama seperti *digital input*, *digital output* hanya dapat mengeluarkan isyarat biner (1 atau 0; High atau Low. Sinyal digital output biasanya digunakan untuk men-drive beban yang besar, seperti relay yang terhubung dengan motor atau fan, power MOSFET yang digunakan untuk menyalakan lampu *High Powered LED* (HPL) dan lain sebagainya.

Seperti yang telah dibahas sebelumnya, Arduino Uno dan Leonardo memiliki 14 pin digital I/O (*input output*). 1 pin hanya dapat digunakan untuk 1 fungsi saja, misalnya pin 13 dikonfigurasi menjadi *digital output* dan tidak bisa dikonfigurasi menjadi *digital input* secara bersamaan.





Sebelum memberi perintah *digital output*, pin harus dikonfigurasi *mode output* pada bagian void(setup) dengan mengetik perintah berikut:

# pinMode(pin, OUTPUT);

Perintah untuk melakukan digital write "high / 1" pada Arduino adalah:

# digitalWrite(pin, HIGH);

Perintah untuk melakukan digital write "low / 0" pada Arduino adalah:

digitalWrite(pin, LOW);



Berikut adalah percobaan *digital output* menggunakan mikrokontroler Arduino MKR1000 :





int ledPin0 = 0; int ledPin0 = 1; int ledPin0 = 2; int ledPin0 = 3; void setup() {

pinMode (ledPin0, OUTPUT); pinMode (ledPin1, OUTPUT); pinMode (ledPin2, OUTPUT); pinMode (ledPin3, OUTPUT); Serial.begin(115200);

# void loop()

```
{
digitalWrite(ledPin0, HIGH);
```

delay(100); digitalWrite(ledPin1, HIGH); delay(100); digitalWrite(ledPin2, HIGH); delay(100); digitalWrite(ledPin3, HIGH); delay(100); digitalWrite(ledPin0, LOW) digitalWrite(ledPin1, LOW); digitalWrite(ledPin2, LOW); digitalWrite(ledPin3, LOW); digitalWrite(ledPin2, HIGH); delay(100);

#### }



Berikut adalah percobaan *analog output* menggunakan mikrokontroler Arduino MRK1000 :





```
int ledPin0 = 2;
int ledPin1 = 3;
int ledPin2 = 4;
int ledPin3 = 5;
int analogValue = 255
void setup()
void loop()
analogWrite(ledPin0, analogValue);
delay(50);
```

```
analogWrite(ledPin1, analogValue);
delay(50);
analogWrite(ledPin2, analogValue);
delay(50);
analogWrite(ledPin3, analogValue);
delay(50);
analogValue = analogValue-5;
if (analogValue < 0)
analogValue = 255;
```



Serial monitor merupakan salah satu *tools* dalam Arduino IDE yang dapat digunakan untuk menampilkan data dari Arduino development board pada layar monitor PC melalui jalur komunikasi UART / USART (USB). Serial monitor ini sangat berguna dalam troubleshooting program dan rangkaian terutama pada rangkaian *input* (sensor) dan *output* (aktuator).

Cara menggunakan serial monitor adalah dengan menambakan perintah berikut:

Serial.begin (baudrate); pada bagian void setup() dan jika ingin menyetak (print) data atau tulisan dapat dengan menggunakan perintah Serial.print() dan Serial.println(). Perbedaan diantara keduanya adalah Serial.print akan menempatkan kursor ke pinggir setelah mencetak data dan Serial.println akan menempatkan kursor ke bawah (membuat baris baru) setelah mencetak data. Pada tanda () Serial.print dan Serial.println dapat diisi variable atau teks dengan tanda kutip. Contoh Serial.print("RDD-tech.com").



1. Mencoba program hello world

```
void setup()
  Serial.begin(115200);
}
void loop()
{
 Serial.print("Arduino IoT Training UNISBA");
 delay(500);
 Serial.println("Hebat! Hebat! Hebat!");
 delay(500);
 Serial.println("HELLO WORLD!");
 delay(1000);
}
```



#### 2. Membuat counter pada serial monitor





```
void setup()
                                       else
 Serial.begin(115200);
                                           buttonPushCounter =
                                       buttonPushCounter;
void loop()
 buttonState = digitalRead(pushButton);
                                         lastButtonState = buttonState;
 if (buttonState != lastButtonState)
                                         Serial.println(buttonPushCounter);
    if (buttonState == HIGH)
      buttonPushCounter++;
```