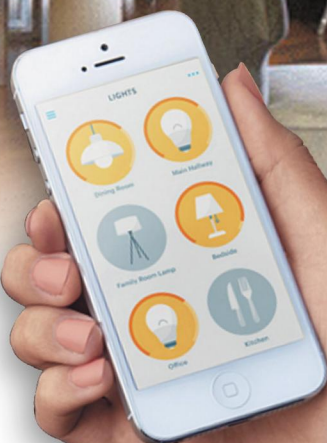


Made Kamisutara, dkk.

Materi  
Buku Ajar



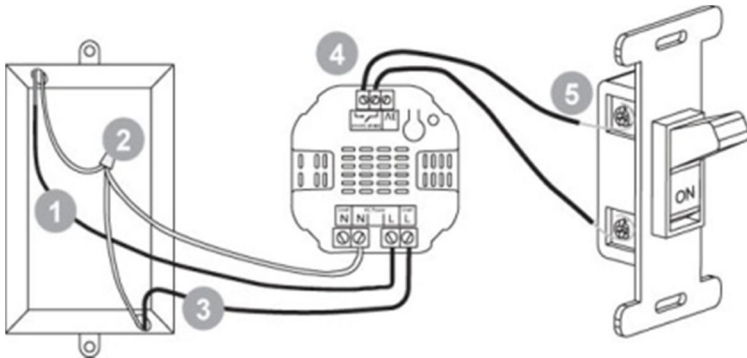
**Micro Controller:**

# WIRELESS SMART SWITCHING

Manajemen Rumah Modern dengan aplikasi desain *switching* yang dikendalikan secara *real-time* melalui sensor memanfaatkan jaringan *wireless* pada produk berteknologi yang terkoneksi melalui jaringan internet

Made Kamisutara, dkk.

Materi Buku Ajar  
Micro Controller:  
**WIRELESS  
SMART SWITCHING**



Diterbitkan oleh  
**Narotama University Press**



**NAROTAMA University Press**

Micro Controller: Wireless Smart Switching  
/disusun oleh Made Kamisutara, dkk.  
... [et al.] ; editor, Seger S.S

**Copyright © 2017** oleh Made Kamisutara

***Hak Cipta Dilindungi Undang-Undang***

*© All Rights Reserved*

Dilarang mengutip dan memperbanyak sebagian atau seluruhnya  
dalam bentuk apapun tanpa ijin tertulis dari Penerbit

**© Cetakan Pertama: Nopember 2017**

Ukuran Buku: A5 (14,8 x 21 cm)

Penyusun : Made Kamisutara  
Slamet Winardi\*  
Tubagus Purworusmiardi  
Editor : Seger, S.S  
Layout/Setting : Gatut Purwantoro  
Design Cover : G. Purwantoro

ISBN : **978-602-6557-05-6**

**© HAK CIPTA DILINDUNGI UNDANG-UNDANG**

Diterbitkan oleh **Narotama University Press**  
Jl. Arief Rachman Hakim No.51 Surabaya 60117  
Telp: 031-5946404, 5995578 Fax: 031-5931213  
Website: [www.narotama.ac.id](http://www.narotama.ac.id)  
Email: [narotamapress@narotama.ac.id](mailto:narotamapress@narotama.ac.id)

# KATA PENGANTAR

Modernisasi memang sangat berpengaruh besar dalam kehidupan masyarakat kita saat ini, dengan adanya era modernisasi yang menggeliat dengan beragam aplikasi yang ditandai dengan semakin berkembangnya arus informasi dan teknologi di setiap sendi kehidupan tak lepas dari teknologi yang senantiasa berevolusi dengan inovasi-inovasi yang baru atau terbaru dan memberikan bermacam kemudahan dalam memenuhi kebutuhan manusia.

Saat ini, produsen berlomba-lomba menawarkan produk baru atau terbaru dengan memberikan layanan kemudahan pada produknya serta *userfriendly* dan bisa dilakukan oleh siapa saja dan dimanapun tanpa terikat jarak dan waktu.

Sebagai contoh bila dulu sebelum kemajuan era digital dan teknologi hadir, untuk mematikan peralatan rumah tangga ataupun lampu kita harus senantiasa berada di tempat, namun sekarang rutinitas semacam itu dapat dilakukan kapan dan dimanapun menggunakan desain *switching* yang dapat dikendalikan secara *real-time* melalui sensor-sensor atau pun memanfaatkan jaringan *wireless* yang terdapat pada beragam produk-produk berteknologi seperti laptop, *smartphone*, dan sejenisnya yang telah terkoneksi dengan internet.

---

Penggunaan sistem *Wireless Smart Switching (WSS)* ini dapat bekerja secara otomatis dengan menggunakan kendali waktu sehingga *User* tidak perlu menekan *switch* manual untuk mematikan dan menghidupkan sebuah lampu tapi sudah secara otomatis akan hidup dan mati berdasarkan waktu yang telah di-*setting*.

Lampu akan menyala atau mati secara otomatis sesuai dengan waktu telah ditentukan dan bila dalam rentang waktu yang ditentukan terjadi perubahan cuaca karena mendung (gelap) sistem akan secara otomatis menghidupkan lampu melalui informasi dari sensor cuaca (LDR) dan akan mematikan kembali bila cuaca kembali terang.

Sistem WSS ini dapat dikendalikan menggunakan *remote control* yang terintegrasi dengan *smartphone* pemakai menggunakan komunikasi via *Bluetooth* dan dapat diakses melalui saluran *wifi* sehingga memungkinkan terhubung dari jarak jauh melalui jaringan internet.

Kepada semua pihak yang telah membantu, baik dari pengumpulan materi, editing naskah, pencetakan dan sebagainya, sehingga penerbitan buku ini dapat terlaksana.

Surabaya, Maret 2017

Penulis

# DAFTAR ISI

<b>KATA PENGANTAR</b> .....	iii
<b>DAFTAR ISI</b> .....	v
<b>BAB I MIKROKONTROLER ARDUINO UNO</b> .....	1
<b>BAB II PROGRAM <i>SOFTWARE</i> ARDUINO</b> .....	5
2.1. Struktur Program Arduino .....	5
2.2. Tipe-tipe Data Arduino .....	7
2.3. Variabel dan Konstanta .....	12
2.4. Operator-operator dalam Program Arduino ..	14
<b>BAB III <i>REAL TIME CLOCK (RTC)</i></b>	
3.1. Diskripsi Umum .....	21
3.2. Manfaat dan Fitur .....	21
3.3. Skematik RTC 3231 .....	22
<b>BAB IV SENSOR CUACA (LDR)</b>	
4.1. Diskripsi Umum .....	27
4.2. Aplikasi LDR .....	29
4.3. Rangkaian dan Modul Uji LDR .....	30
<b>BAB V INFRA MERAH DAN <i>REMOTE CONTROL</i></b>	
5.1. Diskripsi Umum .....	33
5.2. <i>Remote Control</i> .....	34

---

5.3. Modulasi <i>Infrared Remote Control</i> dan Teori <i>Encoding</i> .....	35
5.4. Aplikasi <i>Remote Control</i> Pada Arduino UNO ..	37

## **BAB VI BLUETOOTH DAN SMARTPHONE**

6.1. <i>Bluetooth</i> .....	39
6.2. Cara Kerja <i>Bluetooth</i> .....	40
6.3. Rangkaian Uji <i>Bluetooth</i> dengan Arduino UNO	41
6.4. Menghubungkan <i>Smartphone</i> dengan HC-05 Modul <i>Bluetooth</i> dan Arduino .....	44

## **BAB VII ESP8266 DAN WIFI**

7.1. Modul ESP8266 .....	45
7.2. <i>Wifi</i> .....	48
7.3. Modul Uji ESP8266 Dengan Arduino .....	51

## **BAB VIII SMART SWITCHING**

8.1. Prototipe <i>Wireless Smart Switching</i> .....	60
8.2. Pengujian Waktu (RTC) .....	61
8.3. Pengujian Sensor Cuaca .....	64
8.4. Pengujian Sensor Infra Merah dan <i>Remote</i> <i>Control</i> .....	65
8.5. Pengujian <i>Bluetooth</i> dengan <i>Smartphone</i> .....	67
8.6. Pengujian dengan <i>Wifi</i> .....	68

<b>DAFTAR PUSTAKA</b> .....	73
-----------------------------	----

<b>TENTANG PENULIS</b> .....	75
------------------------------	----

---

## **SANKSI PELANGGARAN PASAL 113**

### **Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta:**

- (1) Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp 100.000.000 (seratus juta rupiah).
  - (2) Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp 500.000.000,00 (lima ratus juta rupiah).
  - (3) Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp 1.000.000.000,00 (satu miliar rupiah).
  - (4) Setiap Orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/atau pidana denda paling banyak Rp 4.000.000.000,00 (empat miliar rupiah).
-



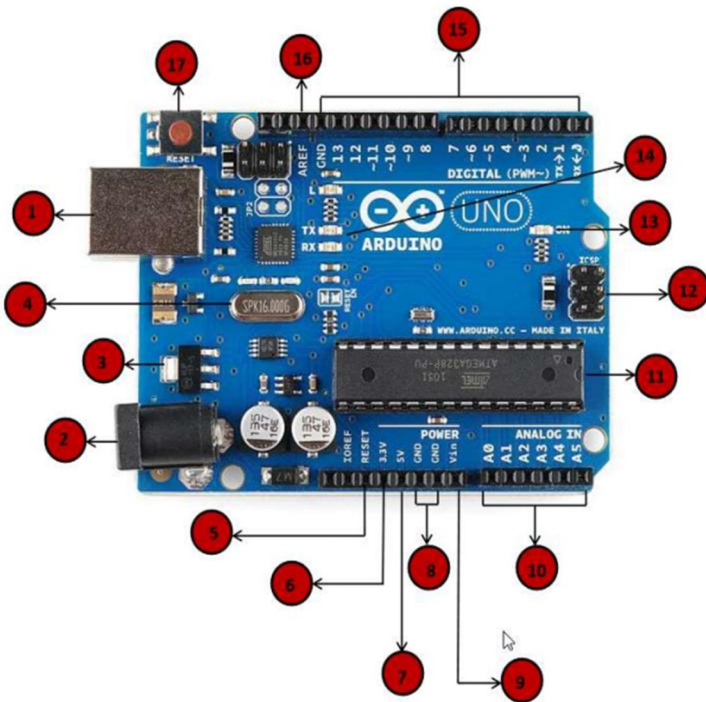


**Universitas Narotama Surabaya**

# BAB I

## MIKROKONTROLER ARDUINO UNO

Dalam bab ini, akan dibahas tentang berbagai komponen pada *board* Arduino. Kita akan mempelajari *board* Arduino UNO karena *board* paling populer di keluarga *board* Arduino. Selain itu, *board* terbaik untuk memulai dengan elektronika dan *coding*. Beberapa *board* terlihat sedikit berbeda dari yang diberikan di bawah ini, tetapi kebanyakan Arduino memiliki mayoritas komponen yang sama.



Gambar 1.1. Board Arduino UNO

1

### **Power USB**

*Board* Arduino dapat disupply dengan menggunakan kabel USB dari komputer. Yang perlu dilakukan adalah menghubungkan kabel USB ke koneksi USB (1)

2

### **Jack Power**

Penghubung ini digunakan untuk memberikan tegangan eksternal sebesar 9- 12 volt dari *power supply*.

3

### **Regulator Tegangan**

Fungsi regulator tegangan untuk mengontrol tegangan diberikan ke *board* Arduino dan menstabilkan tegangan DC yang digunakan oleh prosesor dan komponen lainnya.

4

### **Oscillator Crystal**

Osilator kristal membantu Arduino dalam menangani masalah waktu. Bagaimana Arduino menghitung waktu? Jawabannya adalah, dengan menggunakan osilator kristal. Nilai yang dicetak di atas *body* kristal Arduino adalah SPK16.000G. Ini memberitahu kita bahwa frekuensi 16.000.000 Hertz atau 16 MHz.

5,17

### **Pin Reset**

Fungsi tombol *Reset* pada Arduino yaitu memulai program dari awal. Kita bisa *Reset Board* Arduino UNO dalam dua cara. Pertama, dengan menggunakan tombol *reset* (17) di *board*. Kedua, tombol *reset* eksternal dapat dihubungkan ke pin Arduino berlabel *RESET* (5).

6,7  
8,9

### **Pins (3.3, 5, GND, $V_{in}$ )**

- 3.3V (6): *Supply 3.3 output volt*
- 5V (7): *Supply 5 output volt*

- Sebagian besar komponen yang digunakan pada board Arduino bekerja dengan baik dengan tegangan 3,3 volt dan 5 volt.
- GND (8) (Ground): Ada beberapa pin GND pada Arduino, dimanapun dapat digunakan untuk menghubungkan *ground* dengan rangkaian.
- Vin (9): Pin ini juga dapat digunakan untuk daya pada board Arduino dari sumber daya eksternal, seperti AC pasokan listrik.

10

### Pin-pin Analog

*Board* Arduino UNO memiliki enam pin *input analog* A0 melalui A5. pin ini dapat membaca sinyal dari sensor analog seperti sensor kelembaban atau sensor suhu dan mengkonversi menjadi nilai digital yang dapat dibaca oleh mikroprosesor.

11

### Main Microcontroller

Setiap *board* Arduino memiliki mikrokontroler sendiri (11). Hal ini dapat dianggap sebagai otak dari *board* Arduino UNO. IC utama (*Integrated Circuit*) pada Arduino sedikit berbeda dari *board* yang lain. Mikrokontroler biasanya Perusahaan ATMEL. Informasi ini tersedia di bagian atas IC. Untuk rincian lebih lanjut tentang pembangunan IC dan fungsi, dapat mengacu pada *data sheet*.

12

### Pin ICSP

Sebagian besar, ICSP (12) adalah AVR, *header* pemrograman kecil untuk Arduino terdiri dari Mosi, Miso, SCK, RESET, VCC, dan GND. Hal ini sering disebut sebagai SPI (*Serial Peripheral Interface*), yang dapat dianggap sebagai "perluasan" dari *output*. Pengembangan perangkat *output* ke *master* dari bus SPI.

**13****Indikator *LED Power***

LED ini harus menyala ketika *board* Arduino dipasang ke sumber listrik untuk menunjukkan bahwa *board disupply* dengan benar. Jika lampu ini tidak menyala, maka ada sesuatu yang salah dengan koneksi.

**14****LED TX dan RX**

Di *board* Arduino, akan ditemukan dua label: TX (transmit) dan RX (menerima). Mereka muncul di dua tempat di *board* Arduino UNO. Pertama, pada pin digital 0 dan 1, untuk menunjukkan pin bertanggung jawab untuk komunikasi serial. Kedua, TX dan RX LED (13). LED TX berkedip dengan kecepatan yang berbeda saat mengirim data serial. Kecepatan berkedip tergantung pada *baud rate* yang digunakan oleh *board*. LED RX berkedip selama proses penerimaan.

**15****Digital I/O**

*Board* Arduino UNO memiliki 14 digital pin I/O (15) dimana 6 memberikan PWM (*Pulse Width Modulation*) output. pin ini dapat dikonfigurasi untuk bekerja pin digital sebagai masukan untuk membaca nilai-nilai logika (0 atau 1) atau sebagai pin *output* digital untuk men-dorong modul yang berbeda seperti LED, *relay*, dan lain-lain, pin berlabel "~" dapat digunakan untuk meng-hasilkan PWM.

**16****AREF**

AREF singkatan Analog Referensi. Kadang-kadang, digunakan untuk mengatur referensi eksternal tegangan (antara 0 dan 5 Volt) sebagai batas atas untuk pin *input analog*.

## BAB II

# PROGRAM *SOFTWARE* ARDUINO

### 2.1. STRUKTUR PROGRAM ARDUINO

Dalam bab ini, akan dipelajari secara mendalam, struktur program Arduino dan akan dipelajari lebih terminologi baru yang digunakan dalam dunia Arduino. Perangkat lunak Arduino adalah *open-source*. Sumber kode untuk environment Java yang dirilis dibawah GPL dan C/C ++ mikrokontroler perpustakaan berada di bawah LGPL.

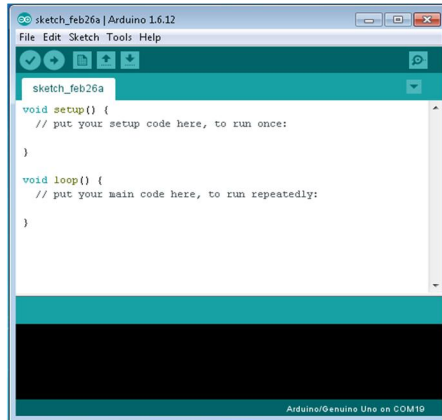
**Sketch:** terminologi baru Yang pertama dalam program Arduino disebut " **Sketch** ".

Struktur program Arduino dapat dibagi dalam tiga bagian utama: **Structure**, **Values** (variabel dan konstanta), dan **Function**. Dalam tutorial ini, akan dipelajari tentang Program *software* Arduino, langkah demi langkah, dan bagaimana dapat menulis program tanpa sintaks atau kesalahan kompilasi.

Mari kita mulai dengan Struktur. struktur perangkat lunak terdiri dari dua fungsi utama yaitu:

- ✓ **Setup( )** function
- ✓ **Loop( )** function

Pada **Gambar 2.1** di bawah ini ditampilkan saat kita masuk pada *software interface* dari program Arduino:



**Gambar 2.1.** Tampilan Awal Software Arduino

```
Void setup ( )  
{  
}
```

**PURPOSE:** fungsi setup () disebut ketika sketch dimulai. digunakan untuk menginisialisasi variabel, mode pin, mulai pmenggunaan *libraries*, dan lain-lain Fungsi pengaturan hanya akan berjalan sekali, setelah setiap power up atau reset board Arduino.

- ✓ INPUT: -
- ✓ OUTPUT: -
- ✓ RETURN: -

```
Void Loop ( )  
{  
}
```

**PURPOSE:** Setelah membuat fungsi setup (), yang menginisialisasi dan menetapkan awal nilai-nilai, fungsi

loop () yang digunakan program untuk mengubah dan menanggapi. Penggunaannya untuk secara aktif dalam mengontrol board Arduino.

- ✓ INPUT: -
- ✓ OUTPUT: -
- ✓ RETURN: -

## 2.2. Tipe-tipe Data Arduino

Tipe data di C mengacu pada sistem yang umum digunakan untuk mendeklarasikan variabel atau fungsi berbagai jenis. Jenis variabel menentukan berapa banyak ruang yang menempati di penyimpanan dan bagaimana pola bit disimpan ditafsirkan. Tabel berikut merupakan jenis data yang akan digunakan selama pemrograman Arduino.

**Tabel 2.1.** Tipe Data Arduino

void	Boolean	char	Unsigned char	byte	int	Unsigned int	word
long	Unsigned long	short	float	double	array	String-char array	String-object

### void

kata kunci **void** hanya digunakan dalam deklarasi fungsi. Hal ini menunjukkan bahwa fungsi ini diharapkan untuk kembali ada informasi untuk fungsi dari yang dipanggil.

Contoh :

```
Void Loop ( )  
{  
    // rest of the code  
}
```



## Boolean

Variabel **Boolean** akan mempunyai satu dari dua nilai, benar atau salah, 0 atau 1. Setiap variabel Boolean menempati satu bit memori.

Contoh:

```
boolean val = false ; // declaration of variable with type boolean and initialize it with false  
  
boolean state = true ; // declaration of variable with type boolean and initialize it with false
```

## Char

Sebuah tipe data yang menggunakan satu byte memori yang menyimpan nilai karakter. Karakter literal ditulis dalam tanda kutip tunggal seperti ini: 'A' dan untuk beberapa karakter, string menggunakan tanda kutip ganda: "ABC".

Namun, karakter yang disimpan sebagai angka. Anda dapat melihat pengkodean tertentu dalam tabel ASCII. Ini artinya untuk melakukan operasi aritmatika pada karakter, dimana nilai ASCII dari karakter yang digunakan. Misalnya, 'A' + 1 sama dengan nilai 66, karena nilai ASCII dari huruf kapital A adalah 65.

Contoh :

```
Char chr_a = 'a' ;//declaration of variable with type char and initialize it with character a  
  
Char chr_c = 97 ;//declaration of variable with type char and initialize it with character 97
```

Tabel 2.2. Kode ASCII

Ascii Chart																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SPC	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
8	€		f	"	...	†	‡	ˆ	%oo	§	<	œ	ž			
9	\	'	"	"	•	—	—	™	š	>	œ	ž	ÿ			
A	i	€	£	¤	¥	¦	§	¨	©	ª	«	¬	­	®	¯	
B	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ
	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

### Unsigned Char

**Unsigned char** adalah tipe data tak bertanda atau bernilai positif yang menempati satu byte memori. Tipe data **unsigned char** mengkodekan angka 0 sampai 255.

Contoh :

```
Unsigned Char chr_y = 121 ; // declaration of variable with type Unsigned
char and initialize it with character y
```

### Byte

Sebuah type data byte menyimpan nilai unsigned 8-bit, dari 0 sampai 255.

Contoh :

```
byte m = 25 ;//declaration of variable with type byte and initialize it with  
25
```

### ***Integer (Int)***

Bilangan integer merupakan data jenis utama untuk penyimpanan nilai. int menyimpan nilai 16-bit (2-byte). Nilai yang dihasilkan antara -32.768 ke 32.767 (nilai minimum  $-2^{15}$  dan nilai maksimum dari  $(2^{15} - 1)$ ).

Ukuran data int bervariasi dari board Arduino Karena, int menyimpan nilai 32-bit (4-byte). Hal ini menghasilkan nilai diantara -2147483648 sampai dengan 2147483647 (minimum nilai  $-2^{31}$  dan nilai maksimum  $(2^{31} - 1)$ ).

Contoh :

```
int counter = 32 ;// declaration of variable with type int and initialize it with  
32
```

### ***Unsigned Int***

*Unsigned int (unsigned integer)* adalah sama seperti int dalam cara yang mereka menyimpan 2 byte nilai. Tidak menyimpan angka negatif, namun, hanya menyimpan nilai-nilai positif, menghasilkan nilai antara 0 - 65.535 ( $2^{16} - 1$ ). Karena tempat menyimpan nilai 4 byte (32-bit), mulai dari 0 sampai 4294967295 ( $2^{32} - 1$ ).

Contoh :

```
Unsigned int counter= 60 ; // declaration of variable with type unsigned int and  
initialize it with 60
```

## **Word**

Di Uno dan board berbasis ATMEGA lainnya, *word* menyimpan sejumlah unsigned 16-bit.

Contoh :

```
word w = 1000 ;//declaration of variable with type word and initialize it with 1000
```

## **Long**

Variabel *long* merupakan variabel ukuran panjang untuk penyimpanan nilai, dan menyimpan 32 bit (4 byte), dari 2147483648 ke 2147483647.

```
Long velocity= 102346 ;//declaration of variable with type Long and initialize it with 102346
```

## **Unsigned Long**

Variabel *unsigned long* merupakan variabel ukuran panjang untuk penyimpanan nilai dalam ukuran 32 bit (4 bytes). *Unsigned long* tidak akan menyimpan angka negative dengan jangkauan nilai dari 0 sampai 4294967295 ( $2^{32} - 1$ ).

Contoh :

```
Unsigned Long velocity = 101006 ;// declaration of variable with type Unsigned Long and initialize it with 101006
```

## **Short**

Tipe data short adalah data 16-bit. Pada semua Arduino (ATMega dan ARM berbasis), sebuah nilai *short* adalah nilai 16-bit (2-byte). Ini mempunyai nilai antara -32.768 sampai dengan 32.767 (nilai minimum  $-2^{15}$  dan nilai maksimum  $(2^{15} - 1)$ ).

Contoh :

```
short val= 13 ;//declaration of variable with type short and initialize it with 13
```

### **Float**

Jenis data untuk angka floating-point adalah angka yang memiliki titik desimal. Nilai floating-point yang sering digunakan untuk mendekati analog dan berkesinambungan nilai-nilai karena memiliki resolusi lebih besar dari bilangan bulat.

Contoh :

```
float num = 1.352; //declaration of variable with type float and initialize it with 1.352
```

### **Double**

Di Uno dan *board* ATMEGA lainnya tipe data *Double* memberikan presisi angka floating-point dan menempati delapan memori byte (8-byte (64 bit)).

Contoh :

```
double num = 45.352 ;// declaration of variable with type double and initialize it with 45.352
```

## **2.3. Variabel dan Konstanta**

Sebelum kita mulai menjelaskan jenis variabel, sebaiknya kita memahami ***variable scope***.

Apa itu Variable scope ?

Variabel dalam bahasa pemrograman C, yang digunakan oleh Arduino, memiliki sifat yang disebut scope. Sebuah scope merupakan bagian dari program dan ada tiga tempat di mana variabel dapat menyatakan:

- ✓ Di dalam fungsi atau blok, yang disebut variabel lokal.
- ✓ Dalam definisi parameter fungsi, yang disebut parameter formal.
- ✓ Di luar semua fungsi, yang disebut variabel global.

## Variabel Lokal

Variabel yang dideklarasikan di dalam fungsi atau blok disebut variabel lokal. Variabel lokal dapat digunakan hanya dengan pernyataan yang berada di dalam fungsi atau blok kode. Variabel lokal tidak dikenali di luar fungsi mereka sendiri. Berikut adalah contoh menggunakan variabel lokal.

```
Void setup ()
{
}
Void loop ()
{
    int x , y ;
    int z ;           Local variable declaration

    x= 0;
    y=0;             actual initialization
    z=10;
}
```

## Variabel Global

Variabel global didefinisikan di luar semua fungsi, biasanya di bagian atas program. Variabel global akan memegang nilai mereka sepanjang program. Sebuah variabel global dapat diakses oleh fungsi apapun. Artinya, variabel global tersedia untuk digunakan di seluruh seluruh program setelah deklarasi.

Contoh berikut ini menggunakan variabel global dan lokal:

```
int T , S ;  
float c =0 ; Global variable declaration  
Void setup ()  
{  
}  
Void loop ()  
{  
    int x , y ;  
    int z ; Local variable declaration  
  
    x= 0;  
    y=0; actual initialization  
    z=10;  
}
```

## 2.4. Operator-operator dalam Program Arduino

Sebuah operator adalah simbol yang memberitahu compiler untuk melakukan proses matematika tertentu atau fungsi logika. Bahasa C banyak menggunakan operator dan menyediakan jenis operator sebagai berikut:

- ✓ Operator Arithmetic
- ✓ Operator Comparison
- ✓ Operator Boolean
- ✓ Operator Bitwise
- ✓ Operator Compound

## Operator Arithmetic

Asumsikan A bernilai 10 dan B bernilai 20

<i>Operator name</i>	<i>Operator simple</i>	<i>Description</i>	<i>Example</i>
<i>assignment operator</i>	=	Memberikan nilai yang sama antara variable sisi kiri dan sisi kanan	A = B
<i>addition</i>	+	Menjumlahkan dua variabel	A+B
<i>subtraction</i>	-	Mengurangkan dua variabel	A-B
<i>multiplication</i>	*	Mengalikan dua buah variabel	A*B
<i>division</i>	/	Membagi dua buah variabel dan hasilnya adalah bilangan bulat	A/B
<i>modulo</i>	%	Mendapatkan sisa hasil bagi dari dua variabel	A%B



## Operator Comparison

Operator name	Operator simple	Description	Example
<i>equal to</i>	<code>==</code>	Memeriksa nilai dari dua operan sama atau tidak, jika ya maka kondisi menjadi benar.	<code>(A == B)</code> jika sama $\rightarrow$ true
<i>not equal to</i>	<code>!=</code>	Memeriksa nilai dari dua operan sama atau tidak, jika tidak maka kondisi menjadi benar.	<code>A!=B</code> jika beda $\rightarrow$ true
<i>less than</i>	<code>&lt;</code>	Memeriksa nilai dari dua operan besar atau kecil, jika B lebih besar A kondisi menjadi benar.	<code>A&lt;B</code> jika B lebih besar dari A $\rightarrow$ true
<i>greater than</i>	<code>&gt;</code>	Memeriksa nilai dari dua operan besar atau kecil, jika A lebih besar B kondisi menjadi benar.	<code>A&gt;B</code> jika A lebih besar dari B $\rightarrow$ true
<i>less than or equal to</i>	<code>&lt;=</code>	Memeriksa nilai dari dua operan Besar, kecil atau sama besar, jika	<code>A&lt;=B</code> jika B lebih besar atau sama

		B lebih besar A dan atau sama besar kondisi menjadi benar.	besar dengan A → true
<i>greater than or equal to</i>	>=	Memeriksa nilai dari dua operan Besar, kecil atau sama besar, jika A lebih besar B dan atau sama besar kondisi menjadi benar.	A>=B jika A lebih besar atau sama besar dengan B → true

### Operator Boolean

<i>Operator name</i>	<i>Operator simple</i>	<i>Description</i>	<i>Example</i>
<i>and</i>	&&	Logika AND akan berniali True bila keduanya true	(A && B) → true
<i>or</i>		Logika Or akan bernilai True bila salah satu atau keduanya true	A    B jika beda → true
<i>not</i>	!	Bernilai kenalikannya jika input true maka output false	!A → false

## Operator Bitwise

<b><i>Operator name</i></b>	<b><i>Operator simple</i></b>	<b><i>Description</i></b>	<b><i>Example</i></b>
<i>and</i>	&	Bitwise AND setiap bit dari dua <i>operand</i>	(A & B)
<i>or</i>		Bitwise OR setiap bit dari dua <i>operand</i>	A B
<i>xor</i>	^	Bitwise XOR setiap bit dari dua <i>operand</i>	A^B
<i>not</i>	~	Membalik setiap bit <i>operand</i>	~A
<i>Shift left</i>	<<	Menggeser ke kiri <i>operand</i> sejumlah dua	A<<2
<i>Shift right</i>	>>	Menggeser ke kanan <i>operand</i> sejumlah dua	A>>2

## Operator Compound

<b><i>Operator name</i></b>	<b><i>Operator simple</i></b>	<b><i>Description</i></b>	<b><i>Example</i></b>
<i>increment</i>	++	Menaikkan satu nilai	A++ atau A=A+1
<i>decrement</i>	--	Menurunkan satu nilai	A-- atau A=A-1

<i>compound addition</i>	$+=$	Menambahkan operand B dengan operand A dan hasilnya disimpan di operand B	$B+=A$ atau $B=B+A$
<i>compound subtraction</i>	$-=$	Mengurangkan operand B dengan operand A dan hasilnya disimpan di operand B	$B-=A$ atau $B=B-A$
<i>compound multiplication</i>	$*=$	Mengalikan operand B dengan operand A dan hasilnya disimpan di operand B	$B*=A$ atau $B=B*A$
<i>compound division</i>	$/=$	Membagikan operand B dengan operand A dan hasilnya disimpan di operand B	$B/=A$ atau $B=B/A$
<i>compound modulo</i>	$\%=$	Mencari sisa bagi operand B dengan operand A dan hasilnya disimpan di operand B	$B\%=A$ atau $B=B\%A$

<i>compound bitwise or</i>	=	Meng-OR-kan A dengan 2 dan hasilnya disimpan di operand A	A  = 2 atau A = A   2
<i>compound bitwise and</i>	&=	Meng-AND-kan A dengan 2 dan hasilnya disimpan di operand A	A &= 2 atau A = A & 2

## **BAB III**

### ***REAL TIME CLOCK (RTC)***

#### **3.1. Diskripsi Umum**

Harga modul DS3231 sangat terjangkau, I2C sangat akurat *real-time clock* (RTC) dengan *temperature compensated* terintegrasi osilator kristal (TCXO) dan kristal. Perangkat menggabungkan input baterai, dan memelihara ketepatan waktu akurat ketika listrik utama ke perangkat terganggu.

Integrasi resonator Kristal meningkatkan akurasi jangka panjang perangkat. Modul DS3231 ini tersedia dalam bentuk komersial dan industri dengan *range* suhu yang berbeda dan dikemas dalam bentuk SO paket dengan 16-pin, dan jarak 300-mil.

RTC menghasilkan detik, menit, jam, hari, tanggal, bulan, dan informasi tahun. Tanggal pada akhir bulan secara otomatis disesuaikan dengan bulan dengan lebih sedikit dari 31 hari, termasuk koreksi untuk tahun kabisat. Jam beroperasi dalam 24 jam atau 12 jam Format dengan AM indikator/PM.

Dua pemrograman alarm dan *programmable output* gelombang persegi yang dihasilkan. Alamat dan data yang ditransfer serial melalui dua jalur bus I2C.

#### **3.2. Manfaat dan Fitur**

RTC dengan akurasi tinggi yang mengatur Semua Fungsi ketepatan waktu.

- ✓ *Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and*

*Year, with Leap-Year Compensation Valid Up to 2100*

- ✓ *Accuracy  $\pm 2\text{ppm}$  from  $0^\circ\text{C}$  to  $+40^\circ\text{C}$*
- ✓ *Accuracy  $\pm 3.5\text{ppm}$  from  $-40^\circ\text{C}$  to  $+85^\circ\text{C}$*
- ✓ *Digital Temp Sensor Output:  $\pm 3^\circ\text{C}$  Accuracy*
- ✓ *Register for Aging Trim*
- ✓ *RST Output/Pushbutton Reset Debounce Input*
- ✓ *Two Time-of-Day Alarms*
- ✓ *Programmable Square-Wave Output Signal*

Interface Serial sederhana untuk menghubungkan banyak mikrokontroler

- ✓ *Fast (400kHz) I2C Interface*

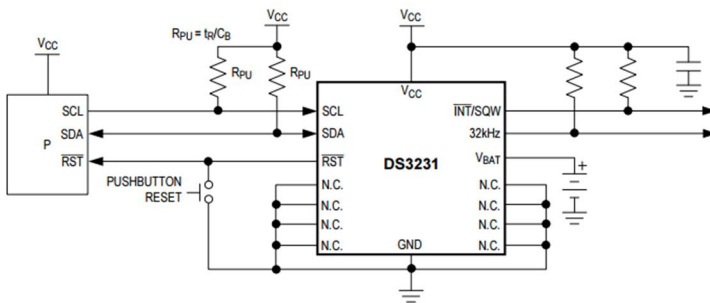
Input baterai Backup untuk Pencatatan Waktu berkelanjutan

- ✓ *Low Power Operation Extends Battery-Backup Run Time*
- ✓ *3.3V Operation*

Operating Temperature Ranges:

- ✓ *Commercial ( $0^\circ\text{C}$  to  $+70^\circ\text{C}$ )*
- ✓ *Industrial ( $-40^\circ\text{C}$  to  $+85^\circ\text{C}$ )*

### 3.3. Skematik RTC 3231



**Gambar 3.1.** Skematik RTC DS3231

## Contoh program untuk menguji rangkaian RTC DS3231

```
#include "Wire.h"
#define DS3231_I2C_ADDRESS 0x68
// Convert normal decimal numbers to binary
coded decimal
byte decToBcd(byte val)
{
    return( (val/10*16) + (val%10) );
}
// Convert binary coded decimal to normal
decimal numbers
byte bcdToDec(byte val)
{
    return( (val/16*10) + (val%16) );
}
void setup()
{
    Wire.begin();
    Serial.begin(9600);
    // set the initial time here:
    // DS3231 seconds, minutes, hours, day, date,
month, year
    // setDS3231time(30,42,21,4,26,11,14);
}
void setDS3231time(byte second, byte minute,
byte hour, byte dayOfWeek, byte
dayOfMonth, byte month, byte year)
{
    // sets time and date data to DS3231
    Wire.beginTransmission(DS3231_I2C_ADDRESS);
    Wire.write(0); // set next input to start at
the seconds register
    Wire.write(decToBcd(second)); // set seconds
    Wire.write(decToBcd(minute)); // set minutes
    Wire.write(decToBcd(hour)); // set hours
    Wire.write(decToBcd(dayOfWeek)); // set day
of week (1=Sunday, 7=Saturday)
    Wire.write(decToBcd(dayOfMonth)); // set date
(1 to 31)
    Wire.write(decToBcd(month)); // set month
    Wire.write(decToBcd(year)); // set year (0 to
99)
```



```

    Wire.endTransmission();
}
void readDS3231time(byte *second,
byte *minute,
byte *hour,
byte *dayOfWeek,
byte *dayOfMonth,
byte *month,
byte *year)
{
    Wire.beginTransmission(DS3231_I2C_ADDRESS);
    Wire.write(0); // set DS3231 register pointer
to 00h
    Wire.endTransmission();
    Wire.requestFrom(DS3231_I2C_ADDRESS, 7);
    // request seven bytes of data from DS3231
starting from register 00h
    *second = bcdToDec(Wire.read() & 0x7f);
    *minute = bcdToDec(Wire.read());
    *hour = bcdToDec(Wire.read() & 0x3f);
    *dayOfWeek = bcdToDec(Wire.read());
    *dayOfMonth = bcdToDec(Wire.read());
    *month = bcdToDec(Wire.read());
    *year = bcdToDec(Wire.read());
}
void displayTime()
{
    byte second, minute, hour, dayOfWeek,
dayOfMonth, month, year;
    // retrieve data from DS3231
    readDS3231time(&second, &minute, &hour,
&dayOfWeek, &dayOfMonth, &month,
&year);
    // send it to the serial monitor
    Serial.print(hour, DEC);
    // convert the byte variable to a decimal
number when displayed
    Serial.print(":");
    if (minute<10)
    {
        Serial.print("0");
    }
    Serial.print(minute, DEC);

```

```

Serial.print(":");
if (second<10)
{
  Serial.print("0");
}
Serial.print(second, DEC);
Serial.print(" ");
Serial.print(dayOfMonth, DEC);
Serial.print("/");
Serial.print(month, DEC);
Serial.print("/");
Serial.print(year, DEC);
Serial.print(" Day of week: ");
switch(dayOfWeek){
case 1:
  Serial.println("Sunday");
  break;
case 2:
  Serial.println("Monday");
  break;
case 3:
  Serial.println("Tuesday");
  break;
case 4:
  Serial.println("Wednesday");
  break;
case 5:
  Serial.println("Thursday");
  break;
case 6:
  Serial.println("Friday");
  break;
case 7:
  Serial.println("Saturday");
  break;
}
}
void loop()
{
  displayTime(); // display the real-time clock
data on the Serial Monitor,
  delay(1000); // every second
}

```



**Universitas Narotama Surabaya**

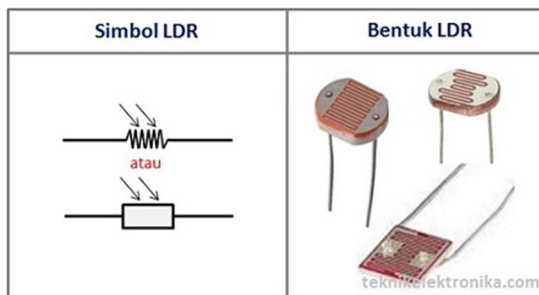
## BAB IV

### SENSOR CUACA (LDR)

#### 4.1. Diskripsi Umum

*Light Dependent Resistor* atau disingkat dengan LDR adalah jenis Resistor yang nilai hambatan atau nilai resistansinya tergantung pada intensitas cahaya yang diterimanya. Nilai Hambatan LDR akan menurun pada saat cahaya terang dan nilai Hambatannya akan menjadi tinggi jika dalam kondisi gelap.

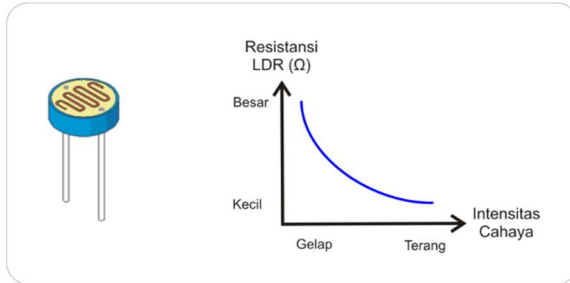
Dengan kata lain, fungsi LDR (*Light Dependent Resistor*) adalah untuk menghantarkan arus listrik jika menerima sejumlah intensitas cahaya (Kondisi Terang) dan menghambat arus listrik dalam kondisi gelap. Naik turunnya nilai Hambatan akan sebanding dengan jumlah cahaya yang diterimanya.



**Gambar 4.1** LDR Light Dependent Resistor

Pada umumnya, Nilai Hambatan LDR akan mencapai 200 Kilo Ohm ( $k\Omega$ ) pada kondisi gelap dan menurun menjadi 500 Ohm ( $\Omega$ ) pada Kondisi Cahaya Terang. LDR

(Light Dependent Resistor) yang merupakan Komponen Elektronika peka cahaya ini sering digunakan atau diaplikasikan dalam Rangkaian Elektronika sebagai sensor pada Lampu Penerang Jalan, Lampu Kamar Tidur, Rangkaian Anti Maling, Shutter Kamera, Alarm dan lain sebagainya.



**Gambar 4.2** Resistansi LDR ketika terkena cahaya

Dengan demikian LDR juga merupakan resistor yang mempunyai koefisien *temperature negative*, dimana resistansinya dipengaruhi oleh intensitas cahaya. LDR terbuat dari Cadium Sulfida, bahan ini dihasilkan dari serbuk keramik.

**Tabel 4.1.** Karakteristik LDR (Light Dependent Resistor)

Parameter	Conditions	Min	Typ	Max	Unit
Cell resistance	1000 LUX	-	400	-	Ohm
	10 LUX	-	9	-	K Ohm
Dark Resistance	-	-	1	-	M Ohm
Dark Capacitance	-	-	3.5	-	pF
Rise Time	1000 LUX	-	2.8	-	ms
	10 LUX	-	18	-	ms
Fall Time	1000 LUX	-	48	-	ms
	10 LUX	-	120	-	ms
Voltage AC/DC Peak		-	-	320	V max
Current		-	-	75	mA max
Power Dissipation				100	mW max
Operating Temperature		-60	-	+75	Deg. C

Biasanya *Cadium Sulfida* disebut juga bahan *photoconductive*, apabila konduktivitas atau resistansi dari *Cadium Sulfida* bervariasi terhadap intensitas cahaya. Jika intensitas cahaya yang diterima rendah maka hambatan juga akan tinggi yang mengakibatkan tegangan yang keluar juga akan tinggi begitu juga sebaliknya di-sinilah mekanisme proses perubahan cahaya menjadi listrik terjadi.

#### **4.2. Aplikasi LDR**

Sel fotokonduktif digunakan dalam berbagai jenis rangkaian dan Aplikasi digital dan analog.

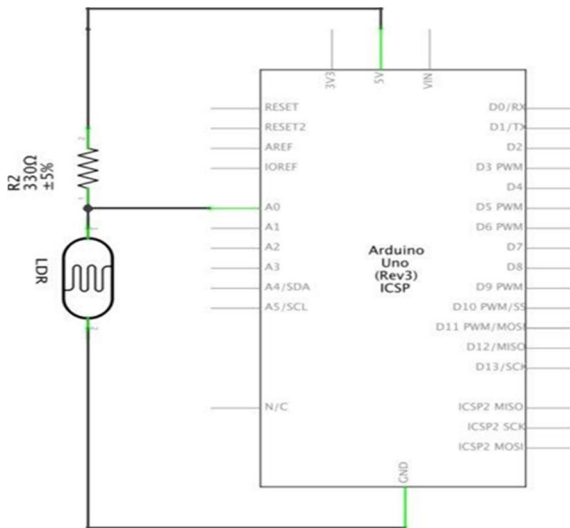
Aplikasi - Aplikasi Analog:

- ✓ *Camera Exposure Control*
- ✓ *Auto Slide Focus - dual cell*
- ✓ *Photocopy Machines - density of toner*
- ✓ *Colorimetric Test Equipment*
- ✓ *Densitometer*
- ✓ *Electronic Scales - dual cell*
- ✓ *Automatic Gain Control—modulated light source*
- ✓ *Automated Rear View Mirror*

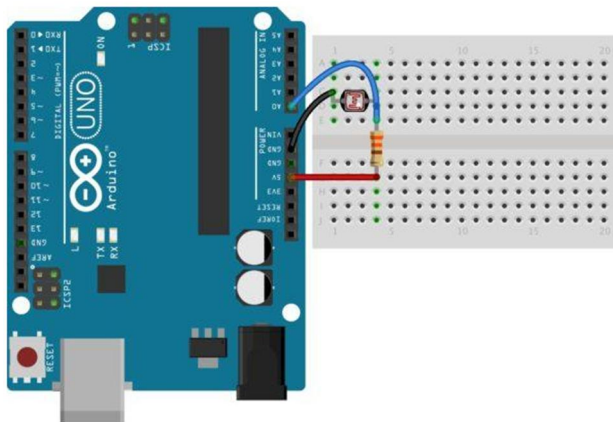
Aplikasi-Aplikasi Digital:

- ✓ *Automatic Headlight Dimmer*
- ✓ *Night Light Control*
- ✓ *Oil Burner Flame Out*
- ✓ *Street Light Control*
- ✓ *Absence / Presence (beam breaker)*
- ✓ *Position Sensor*

### 4.3. Rangkaian dan Modul Uji LDR



**Gambar 4.3.** Rangkaian LDR dan Arduino UNO



**Gambar 4.4.** Modul Uji LDR pada Arduino UNO

## Program Uji Sensor LDR

```
const int pin_ldr = A0;

void setup ()
{
  Serial.begin (9600); //inisialisasi port serial
}

void loop ()
{
  int nilai = analogRead (pin_ldr); //Membaca nilai
  analog dari pin A0
  // Mengkonversi nilai analog tegangan menggunakan
  ADC
  // ADC memiliki resolusi 10bit, sehingga dapat
  mewakili  $2^{10} = 1024$ 
  float tegangan_hasil = 5.0 * nilai / 1024;
  // Catatan: Ini adalah konversi D-to-A
  // Mencetak hasil pada monitor serial
  Serial.print ("Vout =");
  Serial.print (tegangan_hasil);
  Serial.println ("V");
  delay (2000); //jeda selama dua detik
}
```





**Universitas Narotama Surabaya**

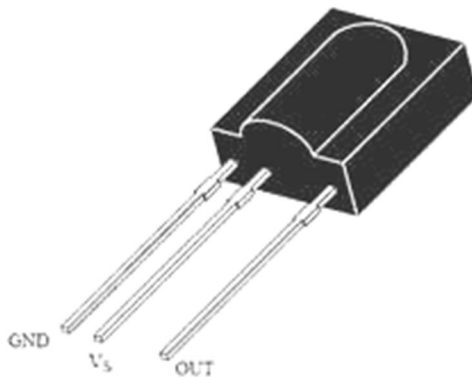
## BAB V

### INFRA MERAH DAN *REMOTE CONTROL*

#### 5.1. Diskripsi Umum

Infra merah adalah sebuah media komunikasi yang menggunakan media cahaya. Penggunaan infra merah sebagai media transmisi data mulai diaplikasikan pada berbagai peralatan seperti televisi, handphone sampai pada transfer data pada PC. Media infra merah ini dapat digunakan baik untuk kontrol aplikasi lain maupun transmisi data.

Cahaya infra merah merupakan cahaya yang tidak tampak. Jika dilihat dengan dengan spektroskop cahaya maka radiasi cahaya infra merah akan Nampak ada spektrum elektromagnet dengan panjang gelombang diatas panjang gelombang cahaya merah. Dengan panjang gelombang ini maka cahaya infra merah ini akan tidak tampak oleh mata namun radiasi panas yang ditimbulkannya masih terasa/dideteksi.



**Gambar 5.1.** Modul Sensor Infra Merah

Sensor ini terdiri dari 2 bagian, yaitu bagian pemancar cahaya dan penerima cahaya. Sebagai media pemancarnya menggunakan infra merah sedangkan penerimanya menggunakan *photodiode*. Untuk rangkaian infra merahnya menggunakan resistor sebagai pembatas arus dan untuk photodiodanya terdiri dari resistor sebagai *pull-up* tegangan.

Photodiode merupakan sensor yang peka terhadap cahaya dan intensitas cahaya yang diterima oleh photodiode akan mempengaruhi nilai resistansinya. Semakin besar intensitas cahaya yang diterima oleh photodiode, maka nilai resistansinya akan semakin kecil dan nilai tegangan *output*-nya akan semakin kecil pula.

Sifat-sifat cahaya infra merah:

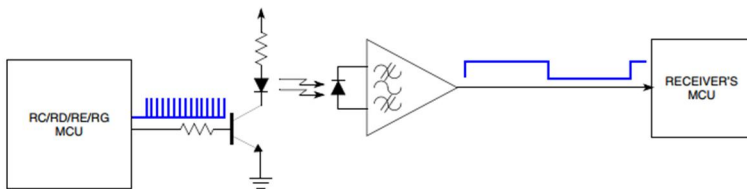
- ✓ Tidak tampak bila dilihat langsung oleh mata manusia.
- ✓ Tidak dapat menembus materi yang tidak tembus pandang.
- ✓ Dapat ditimbulkan oleh komponen yang menghasilkan panas seperti tubuh manusia ataupun tubuh binatang. Berbeda dengan LED biasa, LED infra merah pada penggunaannya dapat diaktifkan dengan:
  - ✓ Tegangan DC
    - Digunakan untuk transmisi jarak dekat.
  - ✓ Tegangan AC ( 30 – 40 KHz )
  - ✓ Digunakan untuk transmisi jarak jauh.

## 5.2. Remote Control

Pernah Anda bertanya-tanya bagaimana inframerah *remote* bekerja? Apa modulasi dan format data yang

berbeda? sederhana ini, murah, namun teknologi yang kuat sangat sering digunakan di rumah hari ini. Catatan aplikasi ini memberikan wawasan beberapa protokol inframerah yang paling sering digunakan dan terutama pelaksanaannya menggunakan Freescale 8-bit MC9S08RC/RD/RE/RG Keluarga mikrokontroler.

Freescale MC9S08RC/RD/RE/RG Keluarga terutama ditargetkan pada orang-aplikasi *remote control* yang dilengkapi dengan CMT kuat (*Carrier Modulator Timer*). Itu CMT modul adalah perangkat khusus yang memungkinkan generasi bentuk gelombang inframerah untuk transmisi sinyal inframerah dengan *overhead software* minimal.



**Gambar 5.2.** Infrared Modulation dan Demodulation

### 5.3. Modulasi *Infrared Remote Control* dan Teori *Encoding*

Cahaya inframerah digunakan dalam aplikasi *remote control* dimodulasi agar penerima dapat membedakan antara sinyal yang diinginkan dan semua sumber kebisingan inframerah.

Ada beberapa perbedaan modulasi dan teknik pengkodean yang digunakan untuk membedakan antara suara yang tidak diinginkan dan sinyal inframerah yang bermanfaat.

Pada dasarnya terdapat 3 (tiga) teknik modulasi yang digunakan yaitu:

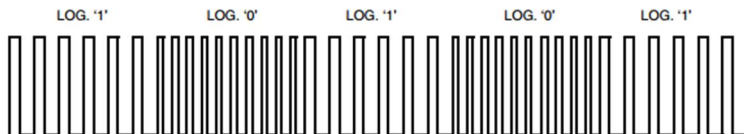
1. *Amplitude Modulation, On-Off Keying, OOK*
2. *FSK, Frekuensi Shift Keying, Frekuensi Modulation*
3. *Flash, 'Pulse' Modulation, Basis Band*

Menggunakan modulasi amplitudo adalah salah satu yang tertua dan paling sederhana teknik, di mana sinyal inframerah membentuk Kelompok pulsa dengan frekuensi tertentu (biasanya 30-60 kHz), dibatasi oleh ruang di mana tidak ada sinyal dihasilkan.

Penerima disetel ke frekuensi tertentu dan semua suara lain tidak akan lewat melalui penerima band pass filter. Receiver terintegrasi dari berbagai produsen yang tersedia (misalnya Infineon, Vishay, Sharp dan lain-lain).

Simple tiga perangkat pin memberikan sinyal yang didemodulasi pada tingkat logika yang sangat mudah untuk interface dengan mikroprosesor penerima. Mereka biasanya diset pada frekuensi tertentu (seperti 30, 33, 36, 38, 40, atau 56 kHz). Amplitudo sistem modulasi menggunakan beberapa metode encoding seperti yang dijelaskan berikut ini.

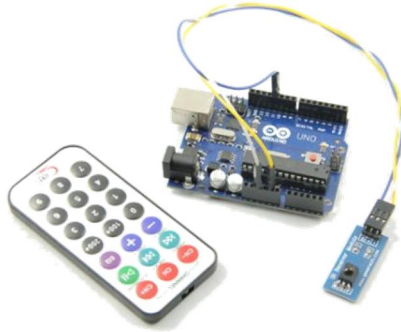
Sebuah frekuensi modulasi dengan menggunakan modulasi frekuensi yang berbeda untuk tingkat logika data. Biasanya ada ruang antara pulsa.



**Gambar 5.3.** Frekuensi Modulasi

#### 5.4. Aplikasi *Remote Control* Pada Arduino UNO

*Remote Control* baru ultra-tipis 38k merupakan inframerah *universal remote control*, *NEC format* pengkodean 21/01-tombol *remote control*, *port USB stereo*, *mobil MP3*, *foot bath*, pencahayaan, frame foto digital, board pengembangan mikrokontroler, dan lain-lain.



**Gambar 5.4.** Rangkaian Uji Infra Red dan *Remote Control*

Fitur yang ditawarkan:

- *Jarak Remote control: lebih 8 meter*
- *Panjang Gelombang infrared: 940Nm*
- *Crystal: frequency oscillation sebesar 455 KHz*
- *IR carrier frequency: 38KHz*
- *Encoding: format encoding adalah NEC, upd6122 encoding scheme, code user 00FF, key coding below picture*
- *Size: 86 \* 40 \* 6mm*
- *Frequency: 38K*
- *Power supply: CR2025/160mAH*
- *Button: free height is less than 3mm, the force 200-350g, the life of more than 200 000*

*Hubungan pin infrared dengan Arduino*

**S -> D11**

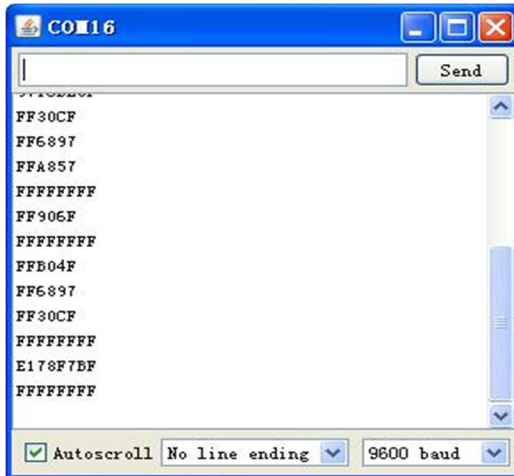
**VCC -> 5V**

## GND -> GND

### Program Uji Remote Control

```
#include <IRremote.h>
int RECV_PIN = 11; //define input pin on Arduino
IRrecv irrecv(RECV_PIN);
decode_results results;
void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
}
void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.value, HEX);
    irrecv.resume(); // Receive the next value
  }
}
```

Ketika menekan tombol apapun pada *remote control*, monitor serial akan menunjukkan kode heksa-desimal yang ada pada tombol *button*. *Every* tombol pada remote control memiliki kode heksadesimal yang sendiri-sendiri. Jika Anda terus menekan tombol apapun, itu menunjukkan FFFFFFFF pada monitor serial.



Gambar 5.5. Hasil Penekanan Tombol Remote Control

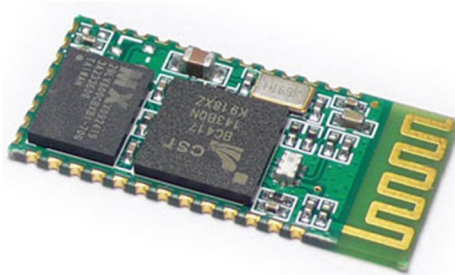
## BAB VI

# BLUETOOTH DAN SMARTPHONE

### 6.1. Bluetooth

*Bluetooth* adalah suatu peralatan media komunikasi yang dapat digunakan untuk menghubungkan sebuah perangkat komunikasi dengan perangkat komunikasi lainnya, *bluetooth* umumnya digunakan di *handphone*, komputer atau PC, tablet, dan lain-lain.

Fungsi *bluetooth* yaitu untuk mempermudah berbagi atau *sharing file*, audio, menggantikan penggunaan kabel dan lain-lain. Saat ini sudah banyak sekali perangkat yang menggunakan *bluetooth*.



Gambar 6.1. Modul Bluetooth

Definisi *h3t* yang lainnya adalah sebuah teknologi komunikasi *wireless* atau tanpa kabel yang beroperasi dalam pita frekuensi 2,4 GHz (antara 2.402 GHz s/d 2.480 GHz) dengan menggunakan sebuah *frequency hopping tranceiver* yang mapu menyediakan layanan komunikasi data dan juga suara secara *real-time* antara *host-host bluetooth* dengan jarak jangkauan layanan yang terbatas.



Pada dasarnya teknologi *bluetooth* ini diciptakan bukan hanya untuk menggantikan atau menghilangkan penggunaan media kabel dalam melakukan pertukaran data atau informasi, tetapi juga mampu menawarkan fitur yang bagus atau baik untuk teknologi *mobile wireless* atau tanpa kabel, dengan biaya yang relatif rendah, konsumsi daya rendah, *interoperability* yang sangat menjanjikan, mudah dalam pengoperasiannya dan juga mampu menyediakan berbagai macam layanan.

## 6.2. Cara Kerja Bluetooth

Sistem *bluetooth* terdiri atas: sebuah radio *transceiver*, *baseband link Management dan Control*, *Baseband (processor core, SRAM, UART, PCM USB Interface)*, *flash dan voice codec*. *Baseband link controller* menghubungkan hardware atau perangkat keras radio ke *baseband processing* dan juga layer protokol fisik. *Link manager* melakukan aktivitas protokol tingkat tinggi, yaitu seperti melakukan *link setup*, autentikasi dan juga konfigurasi.

*Bluetooth* merupakan sistem yang beroperasi atau bekerja pada pita frekuensi antara 2,402 GHz hingga 2,480 GHz. Adanya *bluetooth* mampu memberikan fasilitas kepada Anda untuk melakukan komunikasi atau baik file maupun suara secara real time antara pemilik *bluetooth* itu sendiri. Hanya saja, salah satu *kelemahan bluetooth adalah* jarak jangkauan layanan yang lebih pendek beserta kemampuan transfer yang lebih rendah dibandingkan ketika kita menggunakan kabel.

Kendati demikian, patut diacungi jempol teknologi dari *bluetooth* tersebut. Sistem ini tidak hanya menggantikan kabel saja, namun secara luas, *bluetooth*

mampu menawarkan sebuah fitur yang memakan biaya rendah, fleksibilitas tinggi, sederhana, dan konsumsi daya yang lebih rendah.

Lalu, bagaimana cara kerja *bluetooth* itu sendiri? *Bluetooth* merupakan peralatan yang bekerja dengan melibatkan gelombang radio. Ini bedanya dengan inframerah yang menggunakan gelombang cahaya. Frekuensi jaringan *bluetooth* dibangkitkan oleh daya listrik yang kecil. Oleh sebab itu sebabnya jangkauan *bluetooth* tidaklah terlalu luas, hanya sebatas 10 meter saja.



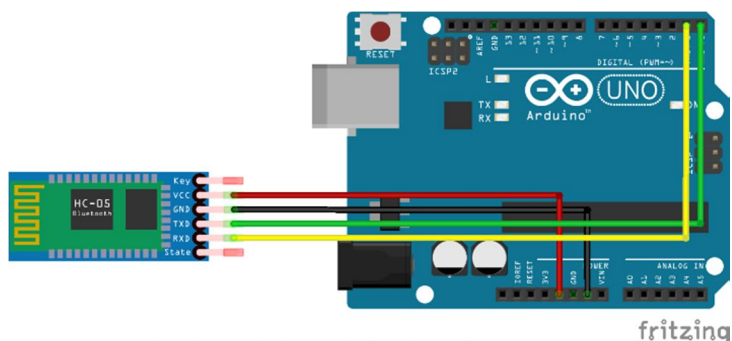
Gambar 6.2. Konektivitas Bluetooth

### 6.3. Rangkaian Uji *Bluetooth* dengan Arduino UNO

Untuk menggunakan modul HC05, cukup menghubungkan VCC untuk output 5V pada Arduino, GND ke *Ground*, RX untuk TX pin Arduino, dan sebaliknya. Jika Anda menggunakan modul BTBee dengan perisai, mengatur *jumper* di *board* sehingga pin DOUT dan pin D0 terhubung dan pin DIN dan D1 juga terhubung.

Hal ini juga dilakukan pada pin RX Arduino Pin 0 dan pin TX Pin 1. Kebebasan untuk menggunakan pin lainnya sebagai RX dan TX Pins, tetapi harus menggunakan **Library**

**SoftwareSerial** dari Arduino untuk mengaktifkannya. Jangan lupa untuk melepas *jumper* saat mengunggah kode untuk Arduino jika Anda telah memilih Pin 0 dan 1 sebagai Serial pin.



**Gambar 6.3.** Hubungan *Bluetooth* dengan Arduino

Jika modul sedang digunakan untuk pertama kalinya, Jika ingin mengubah nama, kode akses dan lain-lain. Untuk melakukan hal ini modul harus diatur ke modus *command*. Menghubungkan pin *Key* untuk setiap pin pada Arduino dan set ke tinggi agar memungkinkan modul untuk diprogram.

Jika menggunakan BTBee, itu sedikit rumit. Sementara *official* wiki mengatakan bahwa *Button Mode* harus diadakan untuk mengubahnya ke *Command Mode*, dan melepaskannya perubahan kembali ke *mode* data, itu tidak terjadi seperti itu: bukan, tahan tombol *Mode*, lalu dengan cepat tekan dan lepaskan Tombol Atur ulang.

Anda akan melihat bahwa Status berkedip lebih lambat dari biasanya LED. Untuk modul program, satu set perintah yang dikenal sebagai AT perintah yang digunakan sebagai berikut:

<i>AT</i>	<i>Check connection status.</i>
<i>AT+NAME ="ModuleName"</i>	<i>Set a name for the device</i>
<i>AT+ADDR</i>	<i>Check MAC Address</i>
<i>AT+UART</i>	<i>Check Baudrate</i>
<i>AT+UART="9600"</i>	<i>Sets Baudrate to 9600</i>
<i>AT+PSWD</i>	<i>Check Default Passcode</i>
<i>AT+PSWD="1234"</i>	<i>Sets Passcode to 1234</i>

### **Program Uji Bluetooth HC-05**

```
#define ledPin 13
int state = 0;

void setup() {
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  Serial.begin(38400); // Default communication
  rate of the Bluetooth module
}

void loop() {
  if(Serial.available() > 0){ // Checks whether
  data is coming from the serial port
  state = Serial.read(); // Reads the data from
  the serial port
}

  if (state == '0') {
    digitalWrite(ledPin, LOW); // Turn LED OFF
    Serial.println("LED: OFF"); // Send back, to
    the phone, the String "LED: ON"
```

```
state = 0;
}
else if (state == '1') {
digitalWrite(ledPin, HIGH);
Serial.println("LED: ON");;
state = 0;
}
}
```

#### **6.4. Menghubungkan *Smartphone* dengan HC-05 Modul *Bluetooth* dan Arduino**

Sekarang disiapkan untuk menghubungkan *smartphone* ke modul *Bluetooth* dan Arduino. Apa yang perlu dilakukan di sini adalah mengaktifkan *Bluetooth* dan *smartphone* akan menemukan modul *Bluetooth* HC-05.

Selanjutnya perlu memasang perangkat dan password default dari modul HC-05 adalah 1234. Setelah dipasangkan perangkat perlu sebuah aplikasi untuk mengontrol Arduino. Ada banyak aplikasi di *Play Store* untuk tujuan ini yang akan bekerja dengan kode Arduino yang kita tulis. Namun, perlu dibuat aplikasi kustom tersendiri untuk menggunakan MIT App Inventor aplikasi *online*.

Berikut aplikasi yang dibuat. Dengan tombol *connect* kita akan menghubungkan *smartphone* ke modul *Bluetooth* dan status teks di bawah tombol akan memberitahu apakah kedua *device* telah berhasil terhubung. Sekarang menggunakan tombol "*Turn ON*" dan "*Turn OFF*" untuk menghidupkan dan mematikan LED. Teks di atas tombol-tombol adalah merupakan isyarat Arduino mengirimkan kembali ke *smartphone* ketika tombol tertentu ditekan.

## BAB VII

### ESP8266 DAN WIFI

#### 7.1. Modul ESP8266

ESP8266 adalah SOC wifi (*System On Chip*) yang diproduksi oleh Espressif Systems. Ini adalah chip yang sangat terintegrasi yang dirancang untuk menyediakan konektivitas internet penuh dalam paket kecil.

ESP8266 dapat digunakan sebagai modul Wifi eksternal, menggunakan standar *AT Command set Firmware* dengan menghubungkannya ke setiap mikro-kontroler menggunakan UART serial, atau langsung berfungsi sebagai *Wifi-enabled* mikro kontroler, dengan pemrograman *firmware* baru menggunakan SDK yang disediakan. Pin GPIO memungkinkan Analog dan Digital IO, ditambah PWM, SPI, I2C, dan lain-lain

Dewasa ini telah hampir satu tahun, dan telah digunakan terutama dalam konteks IOT, dimana digunakan untuk menambahkan konektivitas misalnya untuk proyek Arduino. Sebuah adopsi luas telah difasilitasi oleh harga yang sangat sederhana, mulai 2,50-10 USD tergantung pada fitur yang ditawarkan oleh produsen. Beberapa contoh proyek: Suhu logging dan Web UI, Retro Web Browser, Internet Diaktifkan alarm asap.

**ESP8266** merupakan modul wifi yang berfungsi sebagai perangkat tambahan mikrokontroler seperti **Arduino** agar dapat terhubung langsung dengan wifi dan membuat koneksi TCP/IP.

Modul ini membutuhkan daya sekitar 3.3v dengan memiliki tiga mode wifi yaitu *Station*, *Access Point* dan

*Both* (Keduanya). Modul ini juga dilengkapi dengan prosesor, memori dan GPIO dimana jumlah pin bergantung dengan jenis **ESP8266** yang kita gunakan. Sehingga modul ini bisa berdiri sendiri tanpa menggunakan mikrokontroler apapun karena sudah memiliki perlengkapan layaknya mikrokontroler.

*Firmware default* yang digunakan oleh perangkat ini menggunakan *AT Command*, selain itu ada beberapa Firmware SDK yang digunakan oleh perangkat ini berbasis opensource yang diantaranya adalah sebagai berikut:

- **NodeMCU** dengan menggunakan *basic programming lua*;
- **MicroPython** dengan menggunakan *basic programming python*
- **AT Command** dengan menggunakan perintah-perintah *AT command*

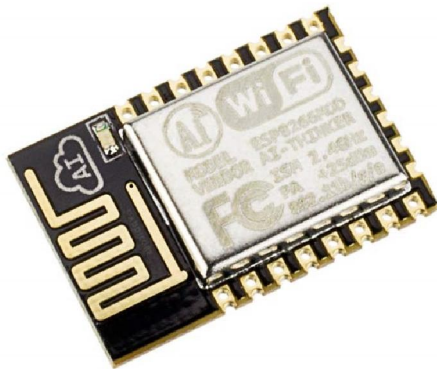
Untuk pemrogramannya sendiri kita bisa menggunakan **ESPlorer** untuk *Firmware* berbasis **NodeMCU** dan menggunakan *putty* sebagai terminal kontrol untuk *AT Command*. Selain itu kita bisa memprogram perangkat ini menggunakan **Arduino IDE**.

Dengan menambahkan **library ESP8266** pada *board manager* kita dapat dengan mudah memprogram dengan *basic program arduino*. Ditambah lagi dengan harga yang cukup terjangkau, Anda dapat membuat berbagai proyek dengan modul ini. Itulah yang melatarbelakangi banyak orang yang menggunakannya modul ini untuk membuat proyek *Internet of Thinking* (IoT).

Berikut ini diuraikan spesifikasi dari Library ESP8266 yang cukup kompleks dan layanan yang dutawarkannya.

## Spesifikasi ESP 8266 / 12

- 802.11 b / g / n
- Wi-Fi Direct (P2P), soft-AP
- Built-in TCP / IP protocol stack
- Built-in TR switch, balun, LNA, power amplifier and matching network
- Built-in PLL, voltage regulator and power management components
- 802.11b mode + 19.5dBm output power
- Built-in temperature sensor
- Support antenna diversity
- off leakage current is less than 10uA
- Built-in low-power 32-bit CPU: can double as an application processor
- SDIO 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU, A-MSDU aggregation and the 0.4 Within wake
- 2ms, connect and transfer data packets
- standby power consumption of less than 1.0mW (DTIM3)



Gambar 7.1. Modul wifi



*Board* ESP8266 tersedia dari beberapa vendor dan dengan *board breakout* yang berbeda. Sebagian besar *board* ESP berbeda dalam jumlah pin yang tersedia, tersedia pula jumlah memori Flash untuk program dan penyimpanan data.

## 7.2. Wifi

**Wi-Fi** merupakan singkatan dari *Wireless Fidelity*, yang memiliki pengertian yaitu sekumpulan standar yang digunakan untuk Jaringan Lokal Nirkabel (*Wireless Local Area Networks – WLAN*) yang didasari pada spesifikasi IEEE 802.11.

Standar terbaru dari spesifikasi 802.11a atau b, seperti 802.11 g, saat ini sedang dalam penyusunan, spesifikasi terbaru tersebut menawarkan banyak peningkatan mulai dari luas cakupan yang lebih jauh hingga kecepatan transfernya.

Awalnya Wi-Fi ditujukan untuk penggunaan perangkat nirkabel dan Jaringan Area Lokal (LAN), namun saat ini lebih banyak digunakan untuk mengakses internet. Hal ini memungkinkan seseorang dengan komputer dengan kartu nirkabel (*wireless card*) atau *Personal Digital Assistant* (PDA) agar terhubung dengan jaringan internet dengan menggunakan titik akses (atau dikenal dengan *hotspot*) terdekat.

Wi-Fi dirancang berdasarkan spesifikasi IEEE 802.11. Sekarang ini terdapat empat variasi dari 802.11, yaitu:

1. 802.11a
2. 802.11b
3. 802.11g
4. 802.11n

Di banyak bagian dunia ini frekuensi yang digunakan oleh Wi-Fi, pengguna tidak diperlukan untuk mendapatkan izin dari pengatur lokal (misal, Komisi Komunikasi Federal di A.S.). 802.11a menggunakan frekuensi yang lebih tinggi dan oleh sebab itu daya jangkauannya lebih sempit, lainnya sama.

Versi Wi-Fi yang paling luas dalam pasarannya di Amerika Serikat sekarang ini (berdasarkan dalam IEEE 802.11b/g) beroperasi pada 2.400 MHz sampai 2.483,50 MHz. Dengan begitu mengizinkan operasi dalam 11 channel (masing-masing 5 MHz), berpusat di frekuensi berikut:

1. Channel 1 – 2,412 MHz;
2. Channel 2 – 2,417 MHz;
3. Channel 3 – 2,422 MHz;
4. Channel 4 – 2,427 MHz;
5. Channel 5 – 2,432 MHz;
6. Channel 6 – 2,437 MHz;
7. Channel 7 – 2,442 MHz;
8. Channel 8 – 2,447 MHz;
9. Channel 9 – 2,452 MHz;
10. Channel 10 – 2,457 MHz;
11. Channel 11 – 2,462 MHz

Secara teknis operasional, Wi-Fi merupakan salah satu varian teknologi komunikasi dan informasi yang bekerja pada jaringan dan perangkat WLAN (*Wireless Local Area Network*). Dengan kata lain, Wi-Fi adalah sertifikasi merek dagang yang diberikan pabrikan kepada perangkat telekomunikasi (internet) yang bekerja di jaringan WLAN dan sudah memenuhi kualitas kapasitas interoperasi yang dipersyaratkan.

Teknologi internet berbasis Wi-Fi dibuat dan dikembangkan sekelompok insinyur Amerika Serikat yang bekerja pada *Institute of Electrical and Electronic Engineers* (IEEE) berdasarkan standar teknis perangkat bernomor 802.11b, 802.11a dan 802.16. Perangkat Wi-Fi sebenarnya tidak hanya mampu bekerja pada jaringan WLAN semata, tetapi juga mampu bekerja di jaringan *Wireless Metropolitan Area Network* (WMAN).

Karena perangkat dengan standar teknis 802.11b diperuntukkan bagi perangkat WLAN yang digunakan di frekuensi 2,4 GHz atau yang lazim disebut frekuensi ISM (*Industrial, Scientific dan Medical*). Sedangkan untuk perangkat yang berstandar teknis 802.11a dan 802.16 diperuntukkan bagi perangkat WMAN atau juga disebut Wi-Max, yang bekerja di sekitar pita frekuensi 5 GHz.

Tingginya animo masyarakat—khususnya di kalangan komunitas Internet—menggunakan teknologi Wi-Fi dikarenakan paling tidak dua faktor. **Pertama**, kemudahan akses. Artinya, para pengguna dalam satu area dapat mengakses Internet secara bersamaan tanpa perlu direpotkan dengan kabel.

Konsekuensinya, pengguna yang ingin melakukan *surfing* atau *browsing* berita dan informasi di Internet, cukup membawa PDA (*Pocket Digital Assistance*) atau laptop berkemampuan Wi-Fi ke tempat dimana terdapat *access point* atau *hotspot*.

Menjamurnya *hotspot* di tempat-tempat tersebut — yang dibangun oleh operator telekomunikasi, penyedia jasa Internet bahkan orang perorangan— dipicu faktor **kedua**, yakni karena biaya pembangunannya yang relatif murah atau hanya berkisar 300 dollar Amerika Serikat.

Peningkatan kuantitas pengguna Internet berbasis teknologi Wi-Fi yang semakin menggejala di berbagai belahan dunia, telah mendorong *Internet Service Providers* (ISP) membangun hotspot di kota-kota besar dunia.

Beberapa pengamat bahkan telah memprediksi pada tahun 2006, akan terdapat *hotspot* sebanyak 800.000 di negara-negara Eropa, 530.000 di Amerika Serikat dan satu juta di negara-negara Asia.

Keseluruhan jumlah penghasilan yang diperoleh Amerika Serikat dan negara-negara Eropa dari bisnis Internet berbasis teknologi Wi-Fi hingga akhir tahun 2003 diperkirakan berjumlah 5.4 triliun dollar Amerika, atau meningkat sebesar 33 milyar dollar Amerika dari tahun 2002 ([www.analysys.com](http://www.analysys.com)).

Adapun Hardware varian wi-fi yang saat ini tersedia di pasaran berupa:

1. *PCI*
2. *USB*
3. *PCMCIA*
4. *Compact Flash*

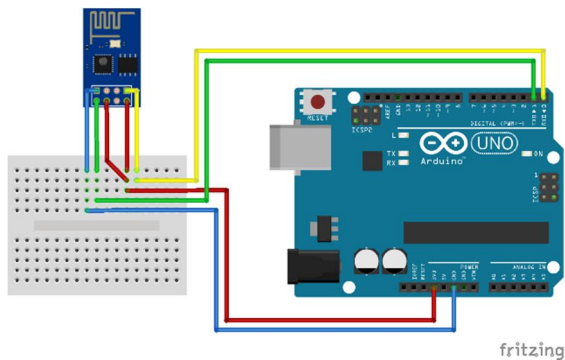
### **7.3. Modul Uji ESP8266 Dengan Arduino**

#### **Spesifikasi Modul ESP8266:**

- *32-bit RISC CPU: Tensilica Xtensa LX106 running at 80 MHz \*\**
- *64 KiB of instruction RAM, 96 KiB of data RAM*
- *External QSPI flash – 512 KiB to 4 MiB\* (up to 16 MiB is supported)*
- *IEEE 802.11 b/g/n Wi-Fi*
- *Integrated TR switch, balun, LNA, power amplifier and matching network*

- WEP or WPA/WPA2 authentication, or open networks
- 16 GPIO pins \*\*
- SPI, I<sup>2</sup>C,
- I<sup>2</sup>S interfaces with DMA (sharing pins with GPIO)
- UART on dedicated pins, plus a transmit-only UART can be enabled on GPIO2
- 1 10-bit ADC

(<http://www.geekstips.com/esp8266-arduino-tutorial-iot-code-example/>)



**Gambar 7.1.** Modul Uji ESP8266 dengan Arduino

Contoh Program Uji untuk mencari sinyal wifi di sekitar modul ESP8266 dengan men-*scan* jaringan Wifi yang tersedia menggunakan shield Wifi. Setiap sepuluh detik, modul ESP8266 scan lagi. Modul ini tidak terhubung ke jaringan apapun, sehingga tidak memerlukan enkripsi.

```
#include "ESP8266WiFi.h"

void setup() {
  Serial.begin(115200);

  // Set WiFi to station mode and disconnect
  // from an AP if it was previously connected
```

```

WiFi.mode(WIFI_STA);
WiFi.disconnect();
delay(100);

Serial.println("Setup done");
}

void loop() {
  Serial.println("scan start");

  // WiFi.scanNetworks will return the number
of networks found
  int n = WiFi.scanNetworks();
  Serial.println("scan done");
  if (n == 0)
    Serial.println("no networks found");
  else
  {
    Serial.print(n);
    Serial.println(" networks found");
    for (int i = 0; i < n; ++i)
    {
      // Print SSID and RSSI for each network
found
      Serial.print(i + 1);
      Serial.print(": ");
      Serial.print(WiFi.SSID(i));
      Serial.print(" (");
      Serial.print(WiFi.RSSI(i));
      Serial.print(")");
      Serial.println((WiFi.encryptionType(i) ==
ENC_TYPE_NONE)?" ":"*");
      delay(10);
    }
  }
  Serial.println("");

  // Wait a bit before scanning again
  delay(5000);
}

```

Contoh modul uji sebagai Server, sketch ini menunjukkan bagaimana untuk menyiapkan Modul ESP8266 sebagai server HTTP secara sederhana.

- \* Server akan menetapkan pin GPIO tergantung pada permintaan
- \* `Http://server_ip/GPIO/0` akan mengatur GPIO2 LOW,
- \* `Http://server_ip/GPIO/1` akan mengatur GPIO2 HIGH
- \* `Server_ip` adalah alamat IP dari modul ESP8266
- \* Dicitak pada layar monitor untuk Serial ketika modul terhubung.

```
#include <ESP8266WiFi.h>

const char* ssid = "your-ssid";
const char* password = "your-password";

// Create an instance of the server
// specify the port to listen on as an argument
WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  delay(10);

  // prepare GPIO2
  pinMode(2, OUTPUT);
  digitalWrite(2, 0);

  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
}
```

```

Serial.println("");
Serial.println("WiFi connected");

// Start the server
server.begin();
Serial.println("Server started");

// Print the IP address
Serial.println(WiFi.localIP());
}

void loop() {
// Check if a client has connected
WiFiClient client = server.available();
if (!client) {
    return;
}

// Wait until the client sends some data
Serial.println("new client");
while(!client.available()){
    delay(1);
}

// Read the first line of the request
String req = client.readStringUntil('\r');
Serial.println(req);
client.flush();

// Match the request
int val;
if (req.indexOf("/gpio/0") != -1)
    val = 0;
else if (req.indexOf("/gpio/1") != -1)
    val = 1;
else {
    Serial.println("invalid request");
    client.stop();
    return;
}
// Set GPIO2 according to the request
digitalWrite(2, val);

```



```

client.flush();
// Prepare the response
String s = "HTTP/1.1 200 OK\r\nContent-Type:
text/html\r\n\r\n<!DOCTYPE
HTML>\r\n<html>\r\nGPIO is now ";
s += (val)?"high":"low";
s += "</html>\n";

// Send the response to the client
client.print(s);
delay(1);
Serial.println("Client disconnected");

// The client will actually be disconnected
// when the function returns and 'client'
object is destroyed
}

```

Contoh uji modul ESP8266 sebagai *client*, *sketch* ini mengirimkan data melalui permintaan HTTP GET ke layanan *data.sparkfun.com* (bisa di-*create* sendiri). Kemudian diperlukan ***streamId*** dan ***PrivateKey*** di *data.sparkfun.com* dan tempelkan dalam program di bawah ini. Atau hanya menyesuaikan *script* ini untuk berbicara dengan server HTTP lainnya.

```

#include <ESP8266WiFi.h>

const char* ssid      = "your-ssid";
const char* password = "your-password";

const char* host = "data.sparkfun.com";
const      char*      streamId          =
".....";
const      char*      privateKey       =
".....";

void setup() {
  Serial.begin(115200);
  delay(10);
}

```

```

// We start by connecting to a WiFi network

Serial.println();
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}

Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}

int value = 0;

void loop() {
    delay(5000);
    ++value;

    Serial.print("connecting to ");
    Serial.println(host);

    // Use WiFiClient class to create TCP
connections
    WiFiClient client;
    const int httpPort = 80;
    if (!client.connect(host, httpPort)) {
        Serial.println("connection failed");
        return;
    }

    // We now create a URI for the request
    String url = "/input/";
    url += streamId;
    url += "?private_key=";
    url += privateKey;

```

```

url += "&value=";
url += value;

Serial.print("Requesting URL: ");
Serial.println(url);

// This will send the request to the server
client.print(String("GET ") + url + "
HTTP/1.1\r\n" +
              "Host: " + host + "\r\n" +
              "Connection: close\r\n\r\n");
unsigned long timeout = millis();
while (client.available() == 0) {
  if (millis() - timeout > 5000) {
    Serial.println(">>> Client Timeout !");
    client.stop();
    return;
  }
}

// Read all the lines of the reply from
server and print them to Serial
while(client.available()){
  String line = client.readStringUntil('\r');
  Serial.print(line);
}

Serial.println();
Serial.println("closing connection");
}

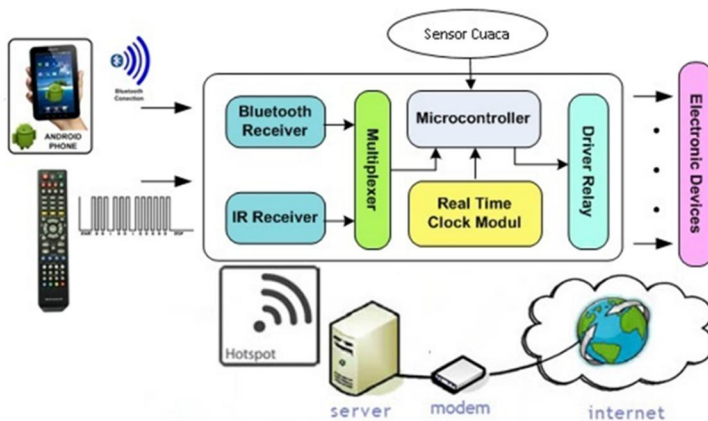
```

## BAB VIII

# SMART SWITCHING

Dalam modul ini secara global terdapat dua tahapan yang harus dikerjakan yaitu desain *hardware* dan desain *software*. Desain *hardware* dilakukan untuk merancang rangkaian sesuai dengan blok diagram dibawah ini sampai dengan pembuatan *Printed Circuit Board* (PCB) sehingga menjadi sebuah produk *wireless smart switching* (WSS).

Sedangkan desain *software* difungsikan untuk menjalankan *hardware* WSS mulai dari pembacaan sensor cuaca, sensor infra merah, sensor *Bluetooth*, membaca *chip* pembangkit waktu secara *real*, serta dapat diakses melalui internet. Berikut ini adalah desain *hardware* *Wireless Smart Switching*.



**Gambar 8.1.** Diagram Blok *Wireless Smart Switching*

## 8.1. Prototipe Wireless Smart Switching

Setelah melakukan perancangan dihasilkan sebuah *prototype Wireless Smart Switching* yang disingkat WSS. Sistem ini akan bekerja berdasarkan sensor dan kontrol melalui *gadget* serta internet. Pertama sistem akan membaca data waktu dari RTC, bila waktu menunjukkan pukul 05.30 maka sistem akan mematikan lampu dan bila waktu menunjukkan pukul 17.30 maka lampu akan otomatis menyala.

Bila diantara waktu 05.30 sampai dengan pukul 17.30 cuaca mendung sekali maka otomatis sistem akan menghidupkan lampu, begitulah sistem otomasi dari *smart switching*. Disamping itu WSS dapat dikendalikan dengan menggunakan *remote control* yang akan mengendalikan sistem dari jarak jauh dengan pancaran sinar infra merah.

Bila seseorang memegang *smartphone* dapat mengendalikan sistem ini dengan menggunakan sinyal *Bluetooth* dengan terlebih dahulu dilakukan *pairing* terhadap sistem dan *smartphone* setelah terjadi kontak maka *smartphone* bisa mengirimkan data untuk mengontrol sistem dan yang terakhir adalah mengakses sistem melalui *web browser*.

Sistem dapat diakses dari tempat yang cukup jauh dengan mengenali IP dari peralatan WSS tersebut. Berikut adalah protipe WSS yang dirancang.



**Gambar 8.2.** Prototipe Wireless Smart Switching

## 8.2. Pengujian Waktu (RTC)

Sistem otomatis WSS ini pertama kali akan membaca seting waktu yang ditetapkan oleh *user*. Seting waktu umumnya untuk menyalakan lampu pada pukul 18.00 dan mematikan lampu pada pukul 05.30.

Sistem akan terus menerus melihat waktu yang dihasilkan oleh modul RTC dengan menggunakan program C Arduino. Program inialisasi untuk modul RTC adalah berikut ini.

```
/****** SETUP rtc *****/  
void setupRTC() {  
  Serial.print("compiled: ");  
  Serial.print(__DATE__);  
  Serial.println(__TIME__);  
  
  Rtc.Begin();  
  // Wire.begin(0, 2); // karena pin yang  
  terbatas, menggunakan pin 0 dan 2 untuk SDA,  
  SCL
```

```

    RtcDateTime compiled = RtcDateTime(__DATE__,
    __TIME__);
    Serial.println();

    if (!Rtc.IsDateTimeValid())
    {
        Serial.println("RTC lost confidence in
the DateTime!");

        Rtc.SetDateTime(compiled);
    }

    if (!Rtc.GetIsRunning())
    {
        Serial.println("RTC was not actively
running, starting now");
        Rtc.SetIsRunning(true);
    }

    RtcDateTime now = Rtc.GetDateTime();
    if (now < compiled)
    {
        Serial.println("RTC is older than compile
time! (Updating DateTime)");
        Rtc.SetDateTime(compiled);
    }
    else if (now > compiled)
    {
        Serial.println("RTC is newer than compile
time. (this is expected)");
    }
    else if (now == compiled)
    {
        Serial.println("RTC is the same as compile
time! (not expected but all is fine)");
    }
    Rtc.Enable32kHzPin(false);

    Rtc.SetSquareWavePin(DS3231SquareWavePin_ModeNone);
}

```

Setelah dilakukan inialisasi terhadap modul RTC kemudian dijalankan program untuk mengakses RTC dengan modul RTC DS\_3231. Program untuk mengakses modul RTC adalah sebagai berikut.

```
/****** program RTC *****/
void runRTC() {
    char timestring[8];
    if (!Rtc.IsDateTimeValid())
    {
        // Common Cuases:
        // 1) the battery on the device is low
        // or even missing and the power line was
        // disconnected
        //Serial.println("RTC lost confidence in
        // the DateTime!");
    }
    RtcDateTime now = Rtc.GetDateTime();
    sprintf(timestring, "%02u:%02u:%02u
    %02/%02%", now.Hour(), now.Minute(),
    now.Second());
    //Serial.print("alarm:\t");
    //Serial.println(timestring);
    if (strstr(timestring, "18:00") != NULL) {
        // Serial.println("alarm");
        flagAlarmRTC = true;
    }
    else if (strstr(timestring, "5:30") != NULL)
    {
        flagAlarmRTC = false;
    }

    u8g.firstPage();
    do {
        draw(timestring);
        u8g.drawStr( 0, 22, "SMART SWITCHING");
    } while ( u8g.nextPage() );
}
```



Hasil pengujian rangkaian modul RTC DS\_3231 adalah sebagai berikut



**Gambar 8.3.** Test Modul RTC DS\_3231

### **8.3. Pengujian Sensor Cuaca**

Cuaca mendung (gelap) akan mempengaruhi sistem WSS ini, dengan memasang sensor LDR pada sistem maka akan dapat membedakan cuaca di sekitar sistem apakah gelap atau terang. Semakin terang cahaya yang diterima maka nilai resistansi dari LDR akan mengecil begitu juga sebaliknya bila cuaca gelap maka nilai resistansi LDR akan membesar.

Berikut adalah cuplikan program untuk menguji sensor cuaca tersebut.

```
void update_lmp() {  
    //Serial.print("timer\n");  
    LDR = map(analogRead(A2), 0, 1023, 100, 0);
```

```

    if (LDR < 60 || statuslampu == true ||
    flagAlaramRTC == true) digitalWrite(lampu,
    HIGH);

    else digitalWrite(lampu, LOW);

    timerflag = true;
}

```

Hasil pengujian dengan sensor cuaca ini disimulasikan dengan menutup sensor dengan tangan, untuk menggambarkan ada atau tidak ada angin.



**Gambar 8.4.** Pengujian Sensor Cuaca (Modul LDR)

#### **8.4. Pengujian Sensor Infra Merah dan Remote Control**

WSS ini juga dapat dikendalikan dengan *remote control* melalui sensor infra merah. WSS ini akan mengenali jenis *remote control* yang digunakan untuk mengendalikan hidup dan matinya sebuah peralatan ataupun lampu penerangan.

Terlebih dahulu WSS ini diprogram dengan kode-kode yang dihasilkan oleh sebuah *remote control* kemudian hasil pembacaan kode-kode *remote control* akan menjalankan instruksi yang diperintahkan oleh mikrokontroler. Tombol yang digunakan untuk menghidupkan

sebuah lampu atau peralatan yaitu sebuah tombol “Power” dan untuk mematikan digunakan tombol yang lain. Berikut ini adalah program untuk *remote control*.

```
///  
// IR remote  
int RECV_PIN = 7;  
IRrecv irrecv(RECV_PIN);  
decode_results results;  
/***** IR RESPONSE *****/  
void IRresponse() {  
  if (irrecv.decode(&results)) {  
    //Serial.println(results.value, HEX);  
    if (results.value == 0xA90) {  
      if (IRupdate + 1000 < millis()) {  
        statuslampu = !statuslampu;  
        IRupdate = millis();  
      }  
    }  
    irrecv.resume(); // Receive the next value  
  }  
}
```

Hasil pengujian dengan menggunakan *remote control* adalah sebagai berikut:



**Gambar 8.5.** Pengujian Dengan Remote Control

## 8.5. Pengujian Bluetooth dengan Smartphone

Pengendalian WSS ini juga dapat dilakukan dengan menggunakan *smartphone* yaitu dengan komunikasi melalui *Bluetooth*. Yang pertama dilakukan adalah melakukan *pairing* terhadap *Bluetooth* yang terpasang pada WSS, setelah *Bluetooth* keduanya terkoneksi maka *smartphone* dapat mengendalikan WSS ini dengan menginstal aplikasi pada *smartphone*.

Dengan menekan tombol secara *touchscreen* pada layar *smartphone* WSS akan bekerja menghidup dan mematikan lampu yang terpasang. Berikut adalah potongan list program untuk mengendalikan WSS lewat *smartphone*.

```
void btresponse() {
  //Serial.print("bt");
  bt.listen();
  String btd = "";
  while (bt.available() > 0) {
    btd = bt.readStringUntil('\n');
  }
  //Serial.print(btd);
  if (btd == "On") {
    statuslampu = !statuslampu;
    // Serial.print(btd);
  }
  btEventFlag = false;
}
```

Hasil pengujian dengan menggunakan *smartphone* adalah sebagai berikut (lihat Gambar 8.6):



**Gambar 8.6.** Pengujian Sensor *Bluetooth* dan *Smartphone*

## 8.6. Pengujian dengan Wifi

WSS ini juga dapat dikendalikan dengan menggunakan jaringan LAN ataupun menggunakan jaringan internet. Untuk dapat mengakses WSS ini diperlukan sebuah *chip* yang dapat diakses melalui akses point.

Dengan menggunakan modul ESP8266 maka sistem WSS ini terkoneksi dengan jaringan LAN melalui sinyal Wifi. Alamat yang digunakan untuk peralatan WSS ini adalah 192.168.4.1.

Dengan Program PHP sederhana untuk mengakses WSS ini lewat komputer dengan koneksi ke jaringan LAN dan menuju ke alamat WSS ini. Setelah wifi terkoneksi WSS ini dapat dikontrol untuk menghidupkan dan mematikan lampu yang terpasang. Berikut adalah potongan program untuk mengendalikan WSS melalui LAN atau internet.

```

/*****/
void InitWifiModule()
{
    sendData("AT+RST\r\n", 2000, DEBUG); // reset

//sendData("AT+CWJAP=\"u8c7i6g5a\", \"sotolamong
an\"\r\n", 2000, DEBUG); //Connect network
    sendData("AT+CWMODE=2\r\n", 3000, DEBUG);
    sendData("AT+CIFSR\r\n", 1000, DEBUG); //
Show IP Adress
    sendData("AT+CIPMUX=1\r\n", 1000, DEBUG); //
Multiple conexions
    sendData("AT+CIPSERVER=1,80\r\n", 1000,
DEBUG); // start comm port 80
}

/*****
***/
// Send AT commands to module
String sendData(String command, const int
timeout, boolean debug)
{
    String response = "";
    esp8266.print(command);
    long int time = millis();
    while ( (time + timeout) > millis())
    {
        while (esp8266.available())
        {
            // The esp has data so display its output
to the serial window
            char c = esp8266.read(); // read the next
character.
            response += c;
        }
    }
    if (debug)
    {
        Serial.print(response);
    }
    return response;
}

```

```

/***** http respon *****/
void httpresponse() {
    esp8266.listen();
    if (esp8266.available()) // check if 8266 is
    sending data
    {
        if (esp8266.find("+IPD,"))
        {
            delay(300);
            int connectionId = esp8266.read() - 48;
            if (esp8266.find("/?on=1")) {
                //digitalWrite(LAMPU, HIGH);
                statuslampu = !statuslampu;
            }

            String webpage = "<html>";
            webpage += "<body align=center><h1>SMART
            SWITCHING</h1></br>";
            int light = LDR;
            webpage += "<h3>Cahaya : ";
            webpage += light;
            webpage += " %</h3>";
            webpage += "<h3>Lampu : ";

            if (statuslampu == true) {
                webpage += "ON";
            } else {
                webpage += "OFF";
            }

            webpage += "</h3><form
            action=\"\"><button name=\"on\" type=\"submit\"
            value=\"1\">ON - OFF</button></form>";
            webpage += "</body></html>";

            String cipSend = "AT+CIPSEND=";
            cipSend += connectionId;
            cipSend += ",";
            cipSend += webpage.length();
            cipSend += "\r\n";

            sendData(cipSend, 1000, DEBUG);
            sendData(webpage, 1000, DEBUG);
        }
    }
}

```

```

String closeCommand = "AT+CIPCLOSE=";
closeCommand += connectionId; // append
connection id
closeCommand += "\r\n";

sendData(closeCommand, 3000, DEBUG);
}
}
espEventFlag = false;
}

```

Hasil pengujian dengan wifi adalah sebagai berikut :



**Gambar 8.6.** Pengndalian dengan Web browser

Berikut pengujian dengan perintah AT *command* untuk menguji sinyal wifi pada peralatan WSS.





**Gambar 8.7.** Perintah AT Command

## DAFTAR PUSTAKA

- Henry G. Barrow, Thura Lin Naing, Robert A. Schneider, Tristan O. Rocheleau, Victor Yeh, Zeying Ren, and Clark T.-C. Nguyen, "A Real-Time 32.768-kHz Clock Oscillator Using a 0.0154-mm<sup>2</sup> Micromechanical Resonator Frequency-Setting Element ", IEEE 2012*
- M. Mock, R. Frings, E. Nett and S. Trikaliotis, "Continuous clock synchronization in wireless real-time applications," Proceedings 19th IEEE Symposium on Reliable Distributed Systems SRDS-2000, Nurnberg, 2000, pp. 125-132. doi: 10.1109/RELDI.2000.885400*
- NajibGhatte, Vinit Pereira, MadurDattaprasad, TusharSurwadkar, "Real-Time Clock Implementation Using I2C Protocol Check ",International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, ISSN-online : 2278 – 8875, Vol. 3, Issue 1, January 2014*
- Real Time Clock, 29 September 2016 (online). Available: [http://en.wikipedia.org/wiki/Real-time\\_clock](http://en.wikipedia.org/wiki/Real-time_clock)*
- Sadeque Reza Khan, Alvir Kabir, Dilshad Ara Hossain, "Designing Smart Multipurpose Digital Clock using Real Time Clock (RTC) and PIC Microcontroller", International Journal of Computer Applications (ISSN: 0975 – 8887), Volume 41– No.9, March 2012*
- Umesh Goyal, Kavita, Neelam Rup, Prakash, "Interfacing Real Time Clock With PIC32 For Real Time Systems", Internatitonal Journal of Engineering Research and Technology (IJERT) ISSN: 2278-0181, vol. 2 Issue 6 Juni 2013*



**Universitas Narotama Surabaya**

## TENTANG PENULIS

**MADE KAMISUTARA** menempuh Sarjana S1 di ITATS lulus tahun 1999 dan melanjutkan S2 di ITS jurusan Teknik Informatika lulus tahun 2007, mulai bergabung di Universitas Narotama Surabaya sejak tahun 2011. Sejak 2000 topik penelitian tentang sistem informasi dan perencanaan strategi di bidang teknologi informasi. Pada tahun 2016 mulai mengembangkan bidang penelitian dengan menggunakan teknologi *hardware* yang sesuai untuk pengembangan Teknologi Informasi.



**SLAMET WINARDI** menempuh Sarjana S1 di ITATS lulus tahun 1997 dan melanjutkan S2 di ITS jurusan Sistem Kontrol lulus tahun 2003, mulai bergabung di Universitas Narotama Surabaya sejak tahun 2011. Sejak 2016 beralih topik penelitian pada dunia *Internet of Things* (IoT) dan aktif memberikan inspirasi kepada mahasiswa tentang perkembangan dan penelitian bidang IoT. Selain itu sebagai praktisi industri di bidang *Mechanical* dan *Electrical*.



**TUBAGUS PURWORUSMIARDI** menempuh Sarjana S1 di Universitas Narotama Surabaya lulus tahun 2006 dan melanjutkan S2 di Universitas Narotama Surabaya Program Studi Magister Manajemen lulus tahun 2015, mulai bergabung di Universitas Narotama Surabaya sejak tahun 2011.





**Universitas Narotama Surabaya**

Di era modernisasi yang tengah kita hadapi saat ini, dimana dalam memenuhi kebutuhan hidupnya manusia berusaha mendapatkannya dengan memprioritaskan pilihannya pada beberapa aspek, terutama pada sisi praktis, mudah, dan murah merupakan kata kunci yang mutlak dipenuhi pada entitas sebuah produk, terlebih dengan adanya beragam aplikasi yang ditandai dengan semakin berkembangnya arus informasi dan teknologi di setiap sendi kehidupan. Keberadaan suatu produk berteknologi yang senantiasa berevolusi dengan inovasi-inovasi baru atau terbarukan sebagai nilai lebih pada produk yang ditawarkan pada konsumen.

Saat ini, produsen berlomba-lomba menawarkan produk baru atau terbarukan dengan memberikan layanan kemudahan pada produknya serta *userfriendly*, bisa dilakukan oleh siapa saja dan dimanapun tanpa terikat jarak dan waktu. Buku ini memberikan sebuah gagasan mengenai konsep rumah modern dan masa depan, sebagai contoh bila dulu sebelum hadirnya era digital dan teknologi, untuk mematikan peralatan rumah tangga ataupun lampu kita senantiasa berada di tempat, namun sekarang rutinitas semacam itu dapat dilakukan kapan dan dimanapun saja menggunakan desain *switching* yang dapat dikendalikan secara *real-time* melalui sensor-sensor memanfaatkan jaringan *wireless* yang terdapat pada beragam produk-produk berteknologi seperti laptop, *smartphone*, dan sejenisnya yang telah terkoneksi dengan internet.

Penggunaan sistem *Wireless Smart Switching* (WSS) ini dapat bekerja secara otomatis dengan menggunakan kendali waktu sehingga *User* tidak perlu menekan *switch* manual untuk mematikan dan menghidupkan sebuah lampu, Lampu akan menyala atau mati secara otomatis sesuai dengan waktu telah ditentukan dan bila dalam rentang waktu yang ditentukan terjadi perubahan cuaca karena mendung (gelap) secara otomatis sistem akan menghidupkan lampu melalui informasi dari sensor cuaca (LDR) dan akan mematikan kembali bila cuaca kembali terang. Sistem WSS ini dapat dikendalikan menggunakan *remote control* yang terintegrasi dengan *smartphone* pemakai dengan menggunakan komunikasi via *Bluetooth* dan dapat diakses melalui saluran wifi sehingga memungkinkan terhubung dari jarak jauh melalui jaringan internet.



**Narotama University Press**

Jl. Arief Rahman Hakim No. 51 Surabaya 60117

Telp. (031) 5946404, 5995578 Fax. (031) 5931213

e-mail: narotamapress@narotama.ac.id

WIRELESS SMART SWITCHING



9 786026 1557056

9 78 - 602 - 6557 - 05 - 6