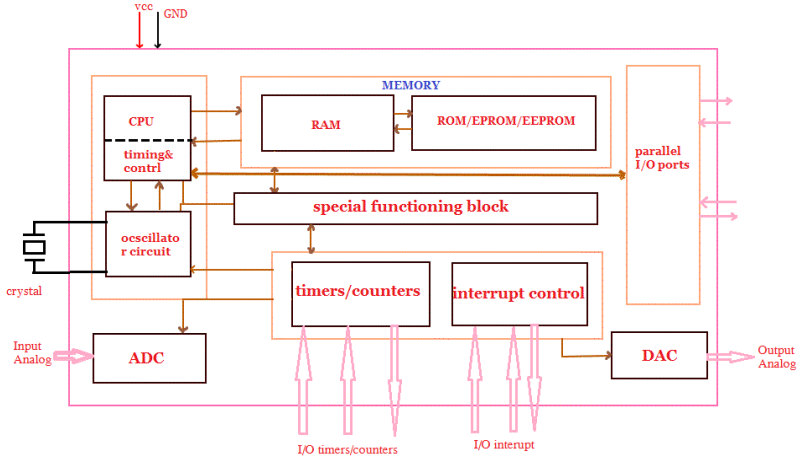
BAB II

LANDASAN TEORI

Landasan teori diperlukan untuk membantu memahami segala teori yang berhubungan dengan pengerjaan alat baik software maupun hardware sehingga tidak terlalu banyak menemukan kesulitan.

2.1 *Microcontroller*

Microcontroller adalah sebuah komputer kecil yang terdapat di dalam sebuah rangkaian terintegrasi. *Microcontroller* pada dasarnya terdiri CPU (*Central Processing Unit*), *memory*, *input/output ports*, *timers and counters*, *interrupt controls*, *analog to digital converters*, *serial interfacing ports* dan *oscillatory circuits*.



Gambar 2.1 Blok Diagram *Microcontroller* [9]

Penjelasan singkat mengenai blok diagram *microcontroller* :

a. CPU (*Central Processing Unit*)

CPU adalah otak dari sebuah *microcontroller*. CPU bertugas untuk mengambil setiap intruksi dalam bentuk kode dan melakukan *decode* (menterjemahkan intruksi) ke dalam bahasa mesin untuk selanjutnya

dilakukan eksekusi. CPU juga bertugas untuk menghubungkan setiap bagian dari *microcontroller* ke dalam sebuah sistem.

b. *Memory*

Fungsi dari sebuah *memory* adalah menyimpan setiap intruksi dan data dari sebuah program. *Microcontroller* biasanya memiliki sejumlah memori seperti SRAM, ROM/EPROM/EEPROM dan *flash memory.*

c. *Parallel Input*/*Output Ports*

Parallel input/*output ports* digunakan untuk melakukan *interface* dengan perangkat lain yang dibutuhkan untuk keperluan sebuah sistem seperti LED, LCD, motor servo, *fingerprint module* dan lain-lain.

d. *Serial Ports*

Serial port digunakan untuk keperluan berbagai *interface* serial antara mikrokontroler dan perangkat lain seperti halnya *parallel port*.

e. *Timers / Counters*

Timers / counters merupakan salah satu fungsi yang sangat berguna dari mikrokontroler. Mikrokontroler memiliki lebih dari satu *timer* dan *counter*. Operasi utama dari *timer* dan *counter* adalah melakukan *clock function* , modulasi, *pulse generations*, pengukuran frekuensi, membuat osilasi, dan lain-lain.

f. ADC (*Analog to Digital Converter*)

ADC berfungsi untuk mengkonversi sinyal analog menjadi digital. Konversi ini dilakukan dengan tujuan mendapatkan output berupa sinyal digital, salah satu contohnya adalah pengukuran suhu yang hasilnya ditampilkan pada sebuah LCD.

g. *Interrupt Control*

Interrupt control digunakan untuk melakukan interupsi pada sebuah program ketika dieksekusi. *Interrupt control* memiliki dua jenis, yaitu *interrupt control internal* (memakai intruksi interupsi) dan *interrupt control eksternal* (memakai akses pin interupsi).

h. *Special Functioning Block*

Special functioning block merupakan bagian tambahan pada sebuah *mikrokontroller* untuk kebutuhan tertentu. Tidak semua *mikrokontroller* menggunakan *special functioning block.*

Salah satu *microcontroller* yang paling banyak digunakan pada jaman sekarang adalah arduino. Arduino adalah sebuah platform komputasi fisik yang bersifat *open-source* berbasis papan/*board microcontroller* sederhana. *Hardware*-nya menggunakan prosesor *atmel* AVR dan *software*-nya memiliki bahasa pemrograman sendiri yaitu C++ arduino. Salah satu alasan kenapa arduino banyak dipakai adalah karena sistem pengoperasiannya yang cukup mudah. Bahasa pemograman yang mudah dipahami dan perangkat tambahan yang mudah didapatkan menjadi daya tarik tersendiri. Arduino memiliki jenis-jenis yang beragam. Jenis-jenis ini memiliki perbedaan dalam segi bentuk, spesifikasi dan performansi, secara singkatnya dijelaskan pada Gambar 2.1 di bawah ini.



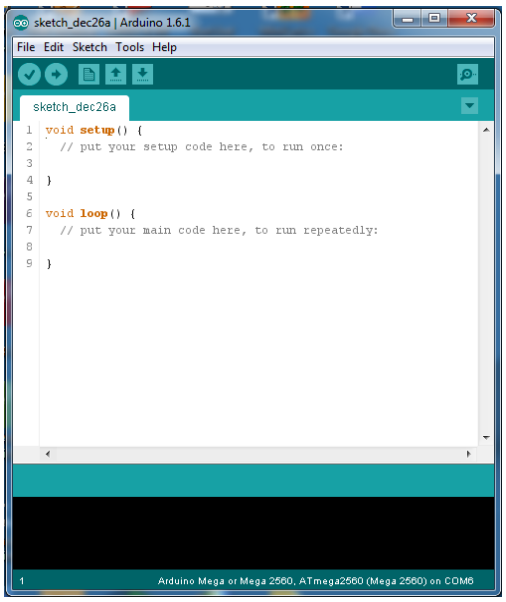
Gambar 2.2 Jenis-jenis Arduino [11]

Tabel 2.1 Perbandingan Jenis-jenis Arduino [11]

	Processor	Processor Voltage	Supply Voltage	Flash	SRAM	Digital I/O Pins	PWM Pins	Analog Inputs	Hardware Serial Ports	Dimensions
Uno	16MHz Atmega 328	5v	7-12v	32Kb	2Kb	14	6	6	1	2.1"x2.7" 53x75mm
Uno Ethernet	16MHz Atmega 328	5v	7-12v	32Kb	2Kb	14	6	6	1	2.1"x2.7" 53x75mm
Mega	16MHz Atmega 2560	5v	7-12v	256Kb	8Kb	54	14	16	4	2.1"x4" 53x102mm
Mega ADK	16MHz Atmega 2560	5v	7-12v	256Kb	8Kb	54	14	16	4	2.1"x4" 53x102mm
Leonardo	16MHz Atmega 32U4	5v	7-12v	32Kb	2.5Kb	20*	7	12*	1	2.1"x2.7" 53x75mm
Due	84MHz ARM SAM3X8E	3.3v	7-12v	512Kb	96Kb	54	12	12	4	2.1"x4" 53x102mm
Micro	16MHz Atmega 32U4	5v	5v	32Kb	2.5Kb	20*	7	12*	1	0.7"x1.9" 18x49mm
Flora	8MHz Atmega 32U4	3.3v	3.5-16v	32Kb	2.5Kb	8*	4	4*	1	1.75" dia 44.5mm dia

2.1.1 Arduino IDE (*Integrated Development Environment*)

Arduino IDE adalah sebuah *software* untuk menulis program, mengedit program sampai mengirim program tersebut ke arduino. Program arduino sering disebut *sketch.*



Gambar 2.3 Arduino IDE

2.1.2 Bahasa Pemograman Arduino

Bahasa Pemograman arduino sering disebut C++ arduino karena struktur pemograman dan tipe datanya mirip dengan C++ pada umumnya. C++ arduino terbagi menjadi 3 bagian utama, *structure*, *function* dan *variables* [1].

1) *Structure*

   a. *Main Structure*

   - void setup( ), nama fungsi yang telah disediakan oleh arduino untuk menyatakan fungsi yang akan dijalankan pertama kali. Fungsi ini berisi kode-kode untuk kepentingan inisialisasi.
   - void loop( ) , fungsi yang secara otomatis dijalankan oleh arduino setelah fungsi *setup* ( ) dieksekusi. Seluruh kode yang ada di fungsi dengan sendirinya akan diulang terus-menerus. Satu-satunya yang bisa menghentikan eksekusi *loop* ( ) adalah berhentinya input catu daya ke papan arduino.

   b. *Control Structures*

   - If….else, format pemogramannya seperti di bawah ini :

     if (kondisi)

     {aksi A}

     else

     {aksi B}

     Jika kondisi *true* maka output akan melakukan aksi A, jika kondisi *false* output akan melakukan aksi B.

   - for, contoh format pemogramannya seperti di bawah ini :

     for (inisialisasi ; kondisi ; penambahan / pengurangan)

contohnya : for (int i=0; i <= 255; i++){analogWrite(PWMpin, i); delay(10); }

loop akan terus berlangsung sampai kondisi terpenuhi, selama loop berlangsung kondisi akan terus dilakukan penambahan/pengurangan.

c. Aturan *Syntax*

Tabel 2.2 Aturan *Syntax* [1]

Syntax	Keterangan
;	Akhir dari setiap *statement*
{}	Batas awal dan akhir program.
//	Memberi komentar pendek
/* */	Memberi komentar panjang
#define	Mendeskripsikan sebuah variabel
#include	Memasukan *library*

d. Operator Aritmatika

Tabel 2.3 Operator Aritmatika [1]

Operator	Keterangan
+	Penjumlahan dan tanda positif.
-	Pengurangan dan tanda negatif
/	Pembagian.
*	Perkalian.
%	Sisa pembagian

e. Operator Pembanding

Tabel 2.4 Operator Pembanding [1]

Operator	Keterangan
==	Operator sama dengan
!=	Operator tidak sama dengan
>	Operator lebih dari
>=	Operator lebih dari atau sama dengan
<	Operator kurang dari

f. Operator *Boolean*

Tabel 2.5 Operator *Boolean* [1]

Operator	Keterangan
&&	Operator “AND”
||	Operator “OR”
!	Operator “NOT”

2) *Function*

a. Digital I/O

Tabel 2.6 Konfigurasi Digital I/O [1]

Digital I/O	Keterangan	Format Program
pinMode( )	Mengkonfigurasi mode pin tertentu pada arduino. Mode yang dipilih bisa *imput* dan *output.*	pinMode(Pin,Mode)
digitalWrite( )	Mengkondisikan *high* atau *low* pada sebuah pin digital.	digitalWrite(Pin,Value)
digitalRead( )	Membaca *high* atau *low* pada sebuah pin digital.	digitalWrite(Pin)

b. Analog I/O

Tabel 2.7 Konfigurasi Analog I/O [1]

Analog I/O	Keterangan	Format Program
analogReference( )	Mengkonfigurasi mo- de tegangan referensi untuk input analog. Mode yang bisa dipilih adalah DEFAULT, INTERNAL, INTERNAL1V1, INTERNAL2V56 dan EXTERNAL	analogReference(Type)
analogWrite()	Menulis nilai analog pada sebuah pin analog.	analogWrite(Pin,Value)
analogRead()	Membaca nilai pada sebuah pin analog.	analogWrite(Pin)

c. Fungsi Matematika

Tabel 2.8 Fungsi Matematika [1]

Fungsi	Keterangan
min(x,y)	Menghitung nilai minimum dari x dan y.
max(x,y)	Menghitung nilai maximum dari x dan y.
abs(x)	Memberikan nilai absolut x.
pow(x,y)	Memberikan nilai balik berupa x^y. Hasilnya mempunyai tipe float.
sin(x)	Memberika nilai balik berupa sinus x. Argumen x dinyatakan dalam radian. Hasilnya mempunyai tipe float.
cos(x)	Memberika nilai balik berupa cosinus x. Argumen x dinyatakan dalam radian. Hasilnya mempunyai tipe float.
tan(x)	Memberika nilai balik berupa tangent x. Argumen x dinyatakan dalam radian. Hasilnya mempunyai tipe float.

3) *Variable*

a. Tipe Data

Tabel 2.9 Tipe Data C++ *Arduino* [1]

Tipe Data	Keterangan	Kebutuhan Memori
boolean	Hanya dapat digunakan untuk menampung dua nilai saja *true* atau *false*	1 byte
char	Berguna untuk menyimpan sebuah nilai karakter seperti ‘A’,’9’ dan ‘*’.	1 byte
unsigned char	Sama dengan tipe char, tetapi bilangan akan dikodekan dalam bentuk bilangan positif antara 0 sampai dengan 255.	1 byte
byte	Berguna untuk menampung bilangan bulat yang berkisar antara 0 sampai 255.	1 byte
int	Berguna untuk menampung bilangan bulat yang berkisar antara -32768 sampai 32767.	2 byte
unsigned nt	Berguna untuk menampung bilang- an bulat yang berkisar antara 0 sampai 65535.	2 byte
word	Tipe word identik dengan usigned int.	2 byte
long	Berguna untuk menampung bilang-an bulat yang berkisar antara -2,147,483, 648 sampai 2,147,483,647	4 byte
unsigned long	Tipe data ini berguna untuk menampung bilang-an bulat yang berkisar antara 0 sampai 4,294,967,295 (2^{23-1})	4 byte
float	Tipe data ini berguna untuk menampung bilang-an real. Angka yang disimpan dari -3,4028235 E+38 sampai 3,4028235 E+38.	4 byte

b. Konversi Tipe Data

Tabel 2.10 Fungsi-fungsi untuk konversi data [1]

Konversi	Keterangan
char( )	Konversi nilai argument tipe char( ) menjadi tipe char.
byte( )	Konversi nilai argument tipe byte( ) menjadi tipe byte.
int( )	Konversi nilai argument tipe int( ) menjadi tipe int.
word( )	Konversi nilai argument tipe word( ) menjadi tipe word.
long( )	Konversi nilai argument tipe long( ) menjadi tipe long.
float( )	Konversi nilai argument tipe float( ) menjadi tipe float.

2.2 Motor DC

Motor DC adalah motor listrik yang memerlukan suplai tegangan arus searah pada kumparan medan untuk diubah menjadi energi gerak mekanik. Kumparan medan pada motor dc disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Motor arus searah, sebagaimana namanya, menggunakan arus langsung yang tidak langsung/direct-unidirectional. Jenis motor dc yang sering digunakan untuk aktuator adalah motor servo dan motor stepper, tetapi untuk sebuah sistem yang membutuhkan aktuator yang membutuhkan *torque* yang besar yang mampu mengangkat beban dalam jarak yang cukup panjang seringkali motor yang digunakan adalah motor *power window*.

2.2.1 Motor *Servo*

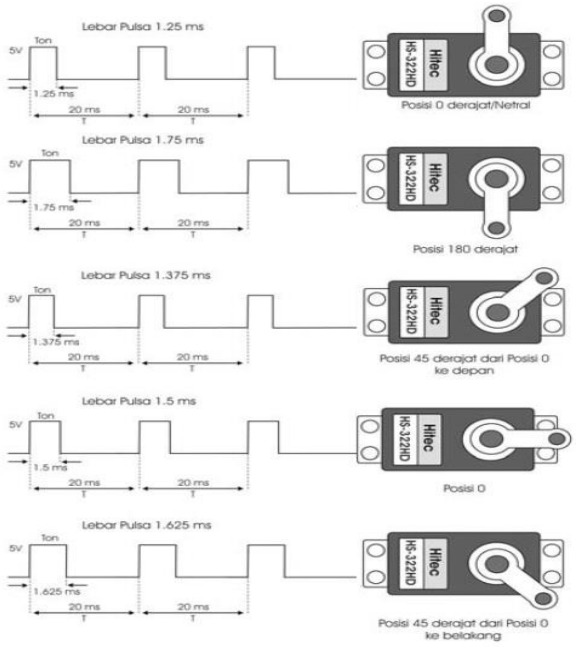
Motor *servo* adalah sebuah motor DC yang dilengkapi rangkaian kendali dengan sistem *closed feedback* yang terintegrasi dalam motor tersebut. Pada motor *servo* posisi putaran sumbu (axis) dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor *servo*. Motor *servo* disusun dari sebuah motor DC, gearbox, variabel resistor (VR) atau potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas maksimum putaran sumbu (axis) motor *servo*. Sedangkan sudut dari sumbu motor *servo* diatur berdasarkan lebar pulsa (PWM) yang pada pin kontrol.



Gambar 2.4 Motor *Servo* [13]

Operasional motor servo dikendalikan oleh sebuah pulsa selebar ± 20 ms, dimana lebar pulsa antara 0.5 ms dan 2 ms menyatakan akhir dari range sudut maksimum. Apabila motor servo diberikan pulsa dengan besar 1.5 ms mencapai gerakan 90°, maka bila kita berikan pulsa kurang dari 1.5 ms maka posisi mendekati 0° dan bila kita berikan pulsa lebih dari 1.5 ms maka posisi mendekati 180°. Motor Servo akan bekerja secara baik jika pada bagian pin kontrolnya diberikan sinyal PWM dengan frekuensi 50 Hz. Dimana pada saat sinyal dengan frekuensi 50 Hz tersebut dicapai pada kondisi Ton duty cycle 1.5 ms, maka rotor

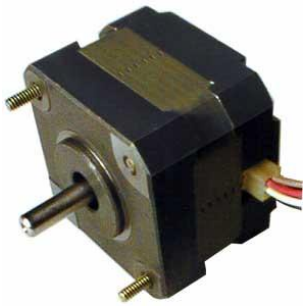
dari motor akan berhenti tepat di tengah-tengah (sudut 0°/ netral). Pada saat Ton *duty cycle* dari sinyal yang diberikan kurang dari 1.5 ms, maka rotor akan berputar ke berlawanan arah jarum jam (*Counter Clock wise*, CCW) dengan membentuk sudut yang besarnya linier terhadap besarnya Ton *duty cycle*, dan akan bertahan diposisi tersebut. Dan sebaliknya, jika Ton *duty cycle* dari sinyal yang diberikan lebih dari 1.5 ms, maka rotor akan berputar searah jarum jam (*Clock Wise*, CW) dengan membentuk sudut yang linier pula terhadap besarnya Ton *duty cycle*, dan bertahan diposisi tersebut.



Gambar 2.5 PWM Motor *Servo* [13]

2.2.2 Motor *Stepper*

Motor *Stepper* adalah suatu motor listrik yang dapat mengubah pulsa listrik yang diberikan menjadi gerakan motor *discret* (terputus) yang disebut *step* (langkah). Satu putaran motor memerlukan 360° dengan jumlah langkah yang tertentu perderajatnya. Ukuran kerja dari motor stepper biasanya diberikan dalam jumlah langkah per-putaran per-detik.



Gambar 2.6 Motor *Stepper* [14]

Motor *stepper* bergerak berdasarkan urutan pulsa yang diberikan kepada motor. Oleh karena itu, untuk menggerakkan motor stepper diperlukan pengendali motor *stepper* yang membangkitkan pulsa-pulsa periodik.

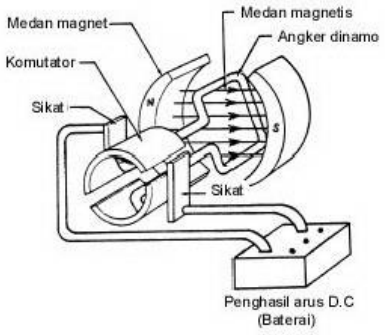
2.2.3 Motor *Power Windows*

Motor *power windows* adalah motor yang merubah energi listrik searah menjadi mekanis yang berupa tenaga penggerak. Motor ini fungsi utamanya adalah untuk sistem *power windows* yang terdapat di dalam mobil, tetapi karena mempunyai torsi tinggi dengan rating tegangan input yang rendah yaitu 12 Volt DC. Motor ini juga memiliki dimensi motor yang relatif *simple* (ramping) dilengkapi dengan *internal gearbox* sehingga memudahkan untuk instalasi mekanik.



Gambar 2.7 Motor *Power Windows* [12]

Prinsip kerja motor *power windows* seperti motor dc pada umumnya, tetapi terdapat perbedaan dalam penambahan *internal gearbox*. Motor *power windows* terdiri dari dua bagian yaitu *stator* (bagian yang diam) berupa magnet permanen dan *rotor* (bagian yang bergerak) berupa koil atau gulungan kawat tembaga. Di setiap ujungnya terhubung dengan komutator, komutator ini dihubungkan dengan kutub positif (+) dan kutub negative (-) dari catu daya. Arus listrik dari kutub positif akan masuk melalui komutator, kemudian berjalan melewati gulungan kawat dan akhirnya kembali ke kutub negatif. Proses ini menyebabkan terjadinya medan elektromagnetik sehingga motor akan bergerak.



Gambar 2.8 Prinsip Kerja Motor *Power Windows* [15]

2.3 Sensor

Sensor adalah komponen elektronika yang berfungsi untuk mengubah besaran mekanis, magnetis, panas, sinar, dan kimia menjadi besarn listrik berupa tegangan, resistansi dan arus listrik. Sensor sering digunakan untuk pendeteksian pada saat melakukan pengukuran atau pengendalian.



Gambar 2.9 Berbagai Contoh Sensor [16]

Secara umum berdasarkan fungsi dan penggunaannya sensor dapat dikelompokan menjadi 3 bagian, yaitu sensor thermal, sensor mekanis dan sensor optik. Tetapi seiring dengan berkembangnya teknologi yang semakin maju ternyata berpengaruh terhadap perkembangan jenis sensor, hal ini ditandai dengan diciptakannya sebuah teknologi yang bernama biometrik. Dari teknologi ini muncul jenis sensor yang baru, yaitu sensor biometrik.

1) Sensor Thermal

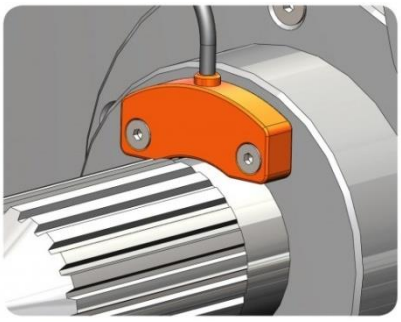
   Sensor thermal atau sensor suhu adalah sensor yang digunakan untuk mendeteksi gejala perubahan panas/temperatur/suhu pada suatu dimensi benda atau dimensi ruang tertentu. Contohnya bimetal, termistor, termokopel, RTD dan lain-lain.



Gambar 2.10 Contoh Sensor Thermal [16]

2) Sensor Mekanik

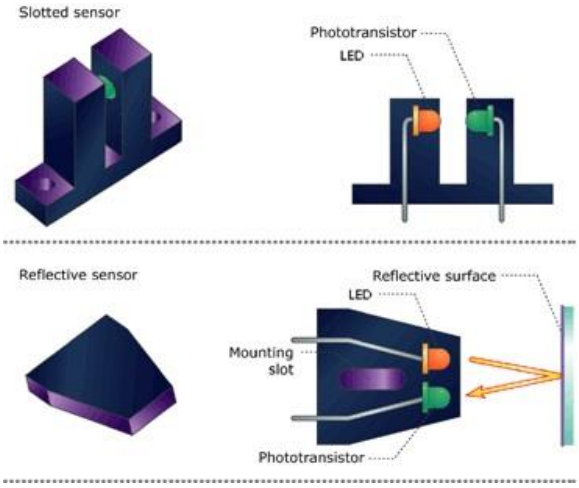
   Sensor mekanik adalah sensor yang mendeteksi perubahan gerak mekanis, seperti perpindahan atau pergeseran posisi, gerak lurus dan melingkar, tekanan, aliran, dan level. Contohnya *strain gage*, *linear variable deferential transformer* (LVDT), *proximity*, *potensiometer*, *load cell*, *bourdon tube , limit switch* dan lain-lain.



Gambar 2.11 Contoh Sensor Mekanik [16]

3) Sensor Optik

Sensor optik atau cahaya adalah sensor yang mendeteksi perubahan cahaya dari sumber cahaya, pantulan cahaya ataupun bias cahaya yang mengernai benda atau ruangan. Contohnya *photo cell*, *photo transistor*, *photo diode*, *photo voltaic*, *photo multiplier*, *pyrometer optic* dan lain-lain.



Gambar 2.12 Contoh Sensor Optik [16]

4) Sensor Biometrik

Sensor biometrik adalah jenis sensor yang digunakan untuk mengukur dan menganalisa pribadi karakteristik, baik fisiologis ataupun perilaku. Karakteristik ini meliputi sidik jari, pola suara, pengukuran tangan, retina mata dan lain-lain. Karakteristik tersebut pada umumnya digunakan untuk memverifikasi identitas untuk sebuah sistem keamanan. Terdapat beberapa

metode diantaranya : *fingerprint scanning, retina scanning*, dan *DNA scanning*. Dua metode terakhir masih dalam taraf penelitian, sedangkan *fingerprint scanning* saat ini telah digunakan secara luas.



Gambar 2.13 Contoh Sensor Biometrik

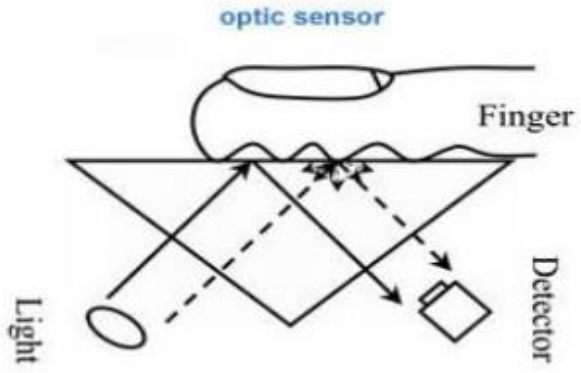
2.3.1 *Fingerprint Module*

Fingerprint Module adalah sensor berbentuk board yang berfungsi mengambil Gambar sidik jari pengguna dan memutuskan apakah pola alur sidik jari dari Gambar yang diambil sama dengan pola alur sidik jari yang ada di *database* modul *fingerprint*.



Gambar 2.14 *Fingerprint Module*

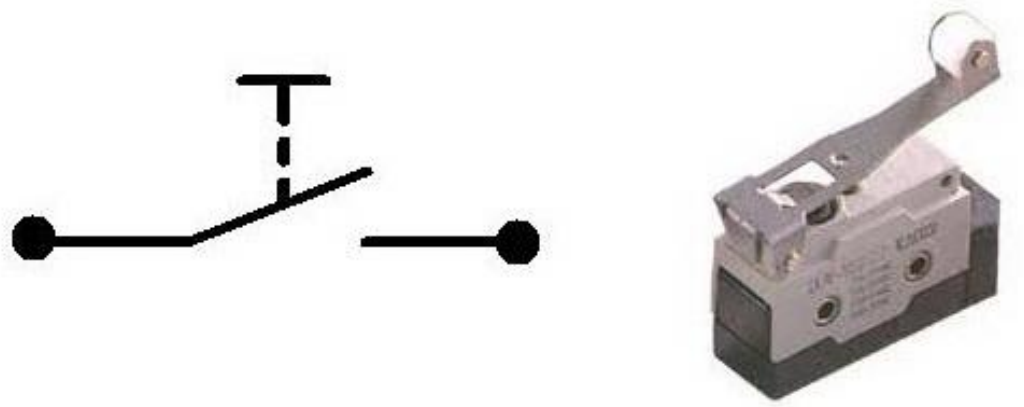
Ada beberapa cara untuk mengambil Gambar sidik jari seseorang, namun salah satu metode yang paling banyak digunakan saat ini adalah *optical scanning* yang juga digunakan pada *fingerprint sensor* yang akan digunakan. Ilustrasi metode *optical scanning* ditunjukan pada Gambar 2.17. Inti dari *scanner optical* adalah c*harge coupled device* (CCD). Proses scan mulai berlangsung saat seseorang meletakkan jari pada lempengan kaca dan sebuah kamera CCD mengambil Gambarnya. *Scanner* memiliki sumber cahaya sendiri, biasanya berupa larik *light emitting diodes* (LED), untuk menyinari alur sidik jarinya.Sistem CCD menghasilkan Gambar jari yang terbalik, area yang lebih gelap merepresentasikan lebih banyakcahaya yang dipantulkan (bagian punggung dari alur sidik jari), dan area yang lebih terang merepresentasikan lebih sedikit cahaya yang dipantulkan (bagian lembah dari alur sidik jari). Sebelum membandingkan Gambar yang baru saja diambil dengan data yang telah disimpan, *processor scanner* memastikan bahwa CCD telah mengambil Gambar yang jelas dengan cara melakukan pengecekan kegelapan pixel rata-rata, dan akan menolak hasil scan jika Gambar yang dihasilkan terlalu gelap atau terlalu terang. Jika Gambar ditolak, *scanner* akan mengatur waktu pencahayaan, kemudian mencoba pengambilan Gambar sekali lagi.



Gambar 2.15 Ilustrasi Metode *Optical Scanning* [5]

2.3.2 *Limit Switch*

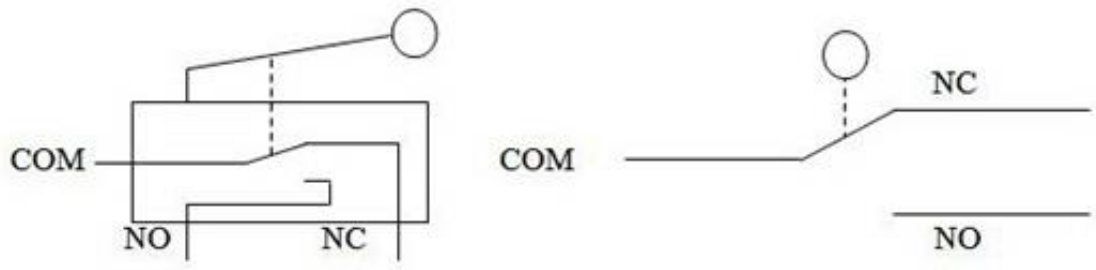
Limit switch merupakan jenis saklar yang dilengkapi dengan katup yang berfungsi menggantikan tombol. Prinsip kerja limit switch sama seperti saklar *Push* ON yaitu hanya akan menghubung pada saat katupnya ditekan pada batas penekanan tertentu yang telah ditentukan dan akan memutus saat saat katup tidak ditekan. *Limit switch* termasuk dalam kategori sensor mekanis yaitu sensor yang akan memberikan perubahan elektrik saat terjadi perubahan mekanik pada sensor tersebut. Penerapan dari *limit switch* adalah sebagai sensor posisi suatu benda (objek) yang bergerak. Simbol *limit switch* ditunjukan pada Gambar 2.18.



Gambar 2.16 Simbol dan Bentuk *Limit Switch* [17]

Limit switch umumnya digunakan untuk memutuskan dan menghubungkan rangkaian menggunakan objek atau benda lain, menghidupkan daya yang besar, dengan sarana yang kecil, sebagai sensor posisi atau kondisi suatu objek. Prinsip kerja *limit switch* diaktifkan dengan penekanan pada tombolnya pada batas/daerah yang telah ditentukan sebelumnya sehingga terjadi pemutusan atau penghubungan rangkaian dari rangkaian tersebut. Limit switch memiliki 2 kontak yaitu NO (*Normally Open*) dan kontak NC (*Normally Close*) dimana salah satu kontak akan

aktif jika tombolnya tertekan. Konstruksi limit switch dapat dilihat seperti Gambar di bawah.



Gambar 2.17 Kontruksi *Limit Switch* [17]

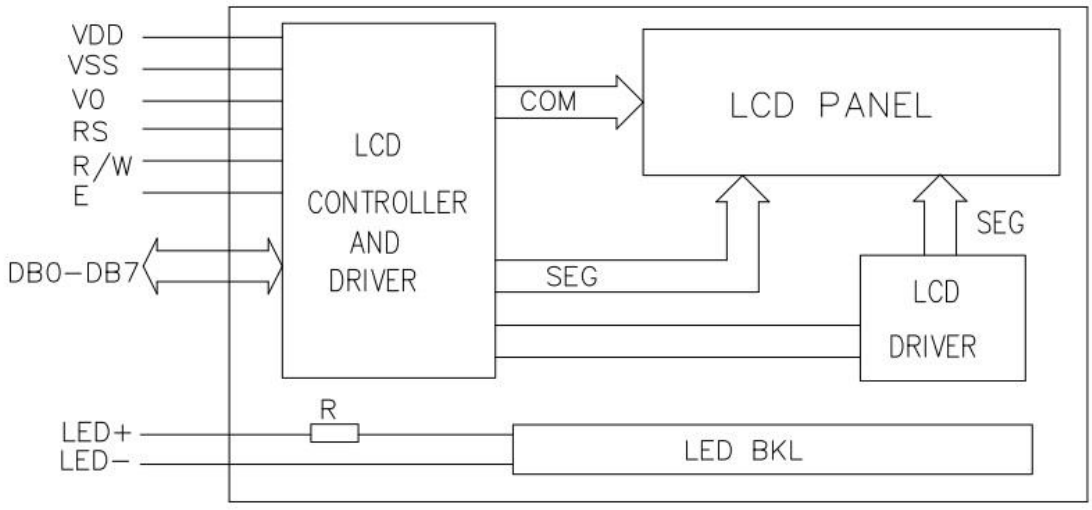
2.4 LCD (*Liquid Crystal Display*)

LCD merupakan perangkat elektronik yang dapat digunakan untuk menampilkan angka atau teks. Ada dua jenis utama layar LCD yang dapat menampilkan numerik (digunakan dalam jam tangan, kalkulator dll) dan menampilkan teks *alfanumerik* (sering digunakan pada mesin foto kopi dan telepon genggam).

Dalam menampilkan numerik ini kristal yang dibentuk menjadi bar, dan dalam menampilkan *alfanumerik* kristal hanya diatur kedalam pola titik. Setiap kristal memiliki sambungan listrik individu sehingga dapat dikontrol secara independen. Ketika kristal *off* (yakni tidak ada arus yang melalui kristal) cahaya kristal terlihat sama dengan bahan latar belakangnya, sehingga kristal tidak dapat terlihat. Namun ketika arus listrik melewati kristal, itu akan merubah bentuk dan menyerap lebih banyak cahaya. Hal ini membuat kristal terlihat lebih gelap dari penglihatan mata manusia sehingga bentuk titik atau bar dapat dilihat dari perbedaan latar belakang.



(a)



(b)

Gambar 2.18 (a) Skematik LCD (b) Blok Diagram LCD [2] [8]

2.4.1 Konfigurasi Pin LCD

Konfigurasi LCD 16x2 dijelaskan secara singkat di dalam tabel 2.10 :

Tabel 2.11 Konfigurasi Pin LCD 16x2 [1]

Pin LCD	Simbol	Keterangan
1	GND	Catu daya, ground
2	VCC	Catu daya, +5V
3	VEE	Pengaturan Kontras LCD
4	RS	*Register Select* • RS=HIGH : untuk mengirim data • RS =LOW : untuk mengirim intruksi
5	R/W	*Read* / *Write Control Bus* • R/W=HIGH : Mode untuk membaca data di LCD • R/W=LOW : Mode penulisan ke LCD • Dihubungkan dengan LOW untuk pengiriman data layar
6	E	*Data Enable*, untuk mengakses LCD. Ketika bernilai LOW, LCD tidak dapat diakses
7 - 14	D0 – D7	Data bus LCD (4 bit atau 8 bit)
15	A	Catu daya layar, positif
16	K	Catu daya layar, negatif

2.4.2 Karakter LCD

Tabel karakter LCD dibawah ini menunjukkan karakter khas yang tersedia pada layar LCD. Kode karakter diperoleh dengan menambahkan angka di atas kolom dengan nomor di sisi baris. Perhatikan bahwa karakter 32-127 selalu sama

untuk semua LCD, tapi karakter 16-31 & 128-255 dapat bervariasi dengan produsen LCD yang berbeda. Oleh karena itu beberapa LCD akan menampilkan karakter yang berbeda dari yang ditunjukkan dalam tabel dibawah ini.

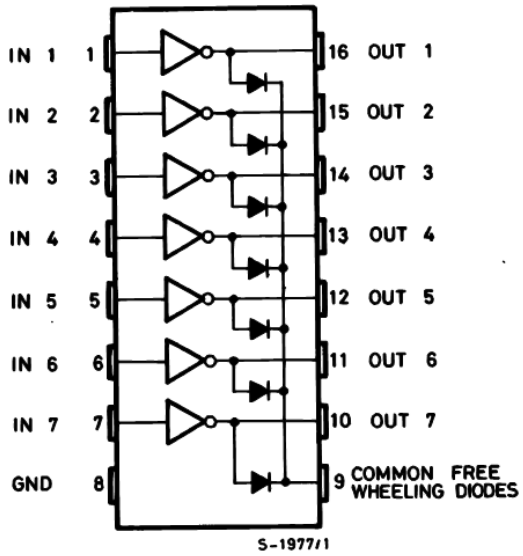
Tabel 2.12 Data Karakter LCD [2]

Upper 4 Bits / Lower 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			0	@	P	`	p				ー	タ	ミ	α	p
xxxx0001	(2)		!	1	A	Q	a	q			。	ア	チ	ム	ä	q
xxxx0010	(3)		"	2	B	R	b	r			「	イ	ツ	メ	ß	θ
xxxx0011	(4)		#	3	C	S	c	s			」	ウ	テ	モ	ε	∞
xxxx0100	(5)		$	4	D	T	d	t			、	エ	ト	ヤ	μ	Ω
xxxx0101	(6)		%	5	E	U	e	u			・	オ	ナ	ユ	σ	ü
xxxx0110	(7)		&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)		'	7	G	W	g	w			ァ	キ	ヌ	ラ	g	π
xxxx1000	(1)		(	8	H	X	h	x			ィ	ク	ネ	リ	√	x̄
xxxx1001	(2)		)	9	I	Y	i	y			ゥ	ケ	ノ	ル	⁻¹	y
xxxx1010	(3)		*	:	J	Z	j	z			ェ	コ	ハ	レ	j	千
xxxx1011	(4)		+	;	K	[	k	{			ォ	サ	ヒ	ロ	ˣ	万
xxxx1100	(5)		,	<	L	¥	l	|			ャ	シ	フ	ワ	¢	円
xxxx1101	(6)		−	=	M	]	m	}			ュ	ス	ヘ	ン	£	÷
xxxx1110	(7)		.	>	N	^	n	→			ョ	セ	ホ	゛	ñ	
xxxx1111	(8)		/	?	O	_	o	←			ッ	ソ	マ	°	ö	■

2.5 ULN2003

ULN2003 adalah sebuah IC tegangan tinggi yang disusun dari 7 chanel transistor darlington, 7 chanel ini terdiri dari 7 pasang pin IC (input dan output). Setiap chanelnya dapat meningkatkan arus masukan sampai 500 mA dan dapat

menahan arus puncak maksimal 600 mA. IC ini biasanya digunakan untuk driver motor, mengatur on/off LED yang berjumlah banyak dan lain-lain. Dalam tugas akhir ini ULN2003 digunakan sebagai driver untuk relay *directional control valve* yang mengatur maju mundurnya piston pendorong.



Gambar 2.19 Konfigurasi Pin ULN2003

2.6 EEPROM

Penempatan atau penyimpanan data dalam arduino dpat dilakukan pada tiga macam memori yaitu memori *flash*, SRAM, dan EEPROM (*Electrically Erasable Programable Read Only Memory*). EEPROM adalah salah satu memori untuk penyimpanan data internal arduino yang sifatnya *non-volatile*, artinya data tidak akan hilang walaupun catudaya arduino mati. Biasanya memori EEPROM diaplikasikan misalnya untuk penyimpanan tabel-tabel data atau konstanta, penyimpanan password dan sebagainya. Dalam tugas akhir ini arduino yang dipakai adalah arduino mega 2560 yang memiliki EEPROM sebesar 4 KB (4096 bytes).

2.7 RTC

Real-time clock disingkat RTC adalah jam di komputer yang umumnya berupa sirkuit terpadu yang berfungsi sebagai pemelihara waktu. RTC umumnya memiliki catu daya terpisah dari catu daya komputer (berupa baterai litium) sehingga dapat tetap berfungsi ketika catu daya komputer terputus. Tipe RTC yang digunakan dalam tugas akhir ini adalah DS3231 yang memiliki spesifikasi sebagai berikut :

1. *Real time clock* (RTC) meyimpan data-data detik, menit, jam, tanggal, bulan, hari dalam seminggu, dan tahun valid hingga 2100.
2. Komunikasi antarmuka serial *two-wire* (I2C).
3. Sinyal keluaran gelombang kotak terprogram (*programmable squarewave*).
4. Ketahanan suhu 0°C hingga 70°C (komersial) dan -40°C hingga +85°C (industrial).
5. Memiliki *crystal oscillator* internal.



Gambar 2.20 RTC DS3231

Penjelasan masing-masing pin RTC DS3231 :

1. 32K, sebagai keluaran frekuensi 32 KHz.
2. SQW, sebagai sinyal kotak (*square wave*) keluaran.
3. SCL, sebagai *serial data clock*.
4. SDA, sebagai *serial data*.
5. VCC, sebagai catu daya positif.
6. GND, sebagai catu daya negatif.

2.8 *Electropneumatic*

Electropneumatic adalah sebuah ilmu yang menggabungkan antara sistem elektro dengan pneumatik (ilmu yang berkaitan dengan gerakan maupun kondisi yang berkaitan dengan udara) . Secara umum *electropneumatic* terdiri dari tiga bagian yaitu, bagian *supply*, bagian *control* dan bagian *actuator*.

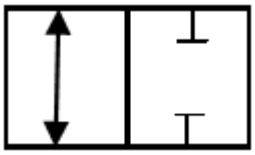
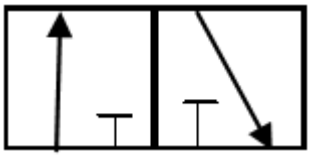
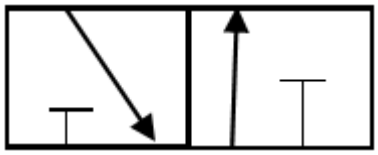
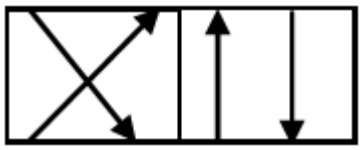
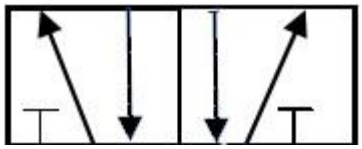
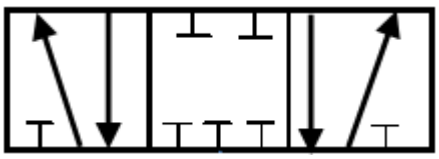
2.8.1 Bagian *Supply*

Supply yang digunakan pada sistem *electropneumatic* adalah kompresor. kompresor merupakan komponen yang berfungsi sebagai penghasil udara bertekanan, bekerja dengan memanfaatkan proses pemampatan udara.

2.8.2 Bagian *Control*

Katup (valve) merupakan element *control* yang digunakan dalam sistem *electropneumatic*. Jenis katup yang digunakan dalam penelitian ini adalah *directional control valve*. Dari namanya sudah bisa diketahui fungsi katup ini adalah untuk mengarahkan dan menutup aliran angin melewati saluran yang telah ditentukan. *Directional control valve* menggunakan relay sebagai pengatur arah alirannya. Pada umumnya terdapat dua relay untuk satu buah DCV. Secara umum DCV terdiri dari 5 tipe, seperti diperlihatkan pada tabel 2.13.

Tabel 2.13. Jenis-jenis *Directional Contol Valve* [7]

Simbol	Keterangan
	Katup 2/2 (2 port 2 posisi)
	Katup 3/2 *Normally Closed* (3 port 2 posisi)
	Katup 3/2 *Normally Open* (4 port 2 posisi)
	Katup 4/2 (4 port 2 posisi)
	Katup 5/2 (5 port 2 posisi)
	Katup 5/3 (5 port 3 posisi)

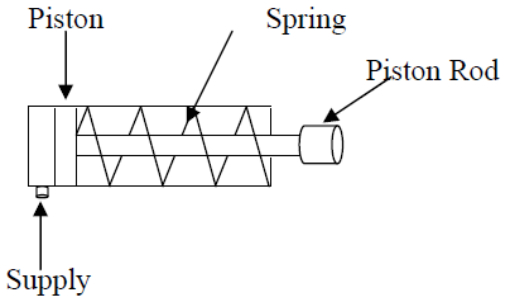
2.8.3 Bagian *Actuator*

Actuator yang paling banyak digunakan dalam sistem otomasi berbasis pneumatic adalah silinder. Silinder bisa digunakan sebagai pengangkat, pendorong, penarik dan lain-lain. Silinder pneumatik kebanyakan berbahan alumunium, hal ini dikarenakan tekanan yang digunakan dalam pengoperasiannya lebih rendah dari silinder hidrolik yang menggunakan bahan besi. Secara umum

silinder pneumatik terdiri dari dua jenis, yaitu *single acting cylinder* dan *double acting cylinder.*

2.8.3.1 *Single Acting Cylinder*

Silinder *single acting* mempunyai spring yang berfungsi sebagai pembalik dari keadaan piston rod yang pada saat tekanan pneumatik tidak aktif, sehinggaa piston akan kembali pada posisi awal. Prinsip kerja dari silinder ini berdasarkan perbedaan gaya yang diterima oleh piston dengan gaya dari spring, yang mana pada saat piston rod maju maka gaya yang diterima oleh piston rod lebih besar dari gaya spring dan pada saat piston rod mundur gaya yang diterima oleh spring lebih besar dari gaya yang diterima oleh piston.

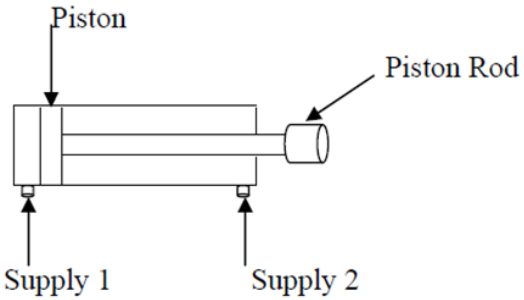


Gambar 2.21. *Single Acting Cylinder* [20]

2.8.3.2 *Double Acting Cylinder*

Silinder double acting memiliki dua saluran input dan setiap inputnya berfungsi sebagai pengendali dari piston, baik pada saat maju ataupun pada saat mundur. Pada saat piston maju input pertama yang berfungsi dan pada saat piston mundur input kedua yang berfungsi. Prinsip kerja utama dari silinder jenis ini tergantung pada gaya yang diterima oleh piston, yang mana pada saat piston rod maju, tekanan yang masuk badalah supply 1 dan memberikan tekanan pada bagian piston yang ada didalam silinder. Pada saat piston rod mundur, tekanan

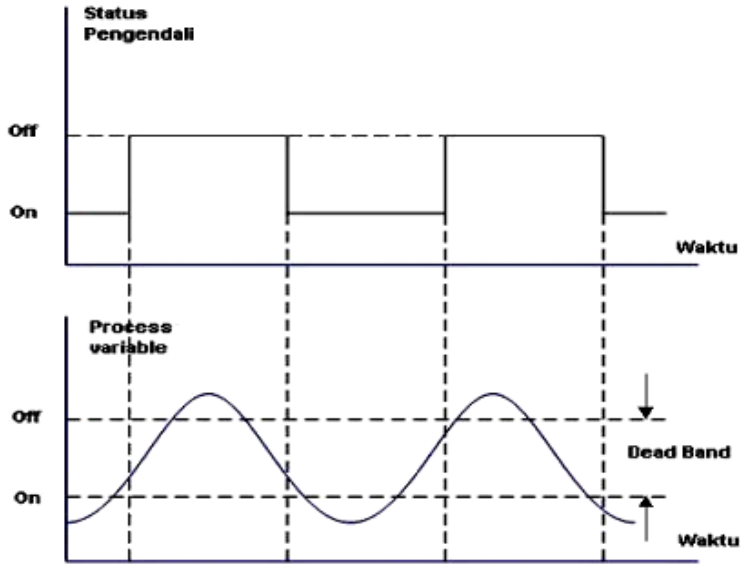
yang masuk adalah supply 2 dan memberikan tekanan pada bagian piston yang ada dalam silinder dan silinder ini tidak ada perbedaaan gaya dalam prinsip kerjanya.



Gambar 2.22. *Double Acting Cylinder* [20]

2.9 Sistem Kendali *On-Off*

Kendali *On-Off* adalah salah satu sistem kendali yang sering dijumpai dalam sebuah pengontrolan. Dalam sistem kendali *on-off*, elemen pembangkit hanya mempunyai dua keadaan yaitu *on* dan *off*. Karena kerjanya yang *on-off*, hasil pengendalian kendali *on-off* akan menyebabkan *process variable* (besaran parameter proses yang dikendalikan) yang bergelombang seperti yang ditunjukan pada gambar di bawah ini.



Gambar 2.23 Kerja Kendali *On-Off* dengan *Dead Band* [3]

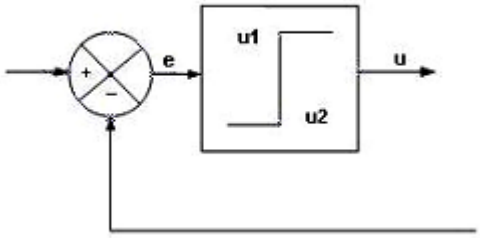
Misal sinyal keluaran kontroler adalah u(t) dan sinyal pembangkit kesalahan (*dead band*) adalah e(t). Pada kendali *on-off*, sinyal u(t) akan tetap pada salah satu nilai maksimum atau minimum tergantung pada sinyal pembangkit kesalahan positif atau negatif sedemikian rupa sehingga [3] :

$u(t) = u1$ untuk $e(t) > 0$

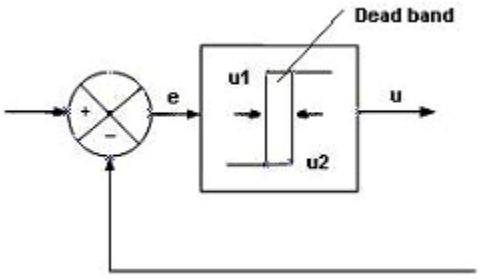
$u(t) = u2$ untuk $e(t) < 0$

dimana

$u1$ dan $u2$ adalah konstan. Harga minimum $u2$ biasanya nol atau $-u1$



Gambar 2.24 Diagram Blok Kendali *on-off* [3]



Gambar 2.25 Diagram Blok Kendali *on-off* dengan *Dead band* [3]

Gambar 2.24 dan 2.25 menunjukan diagram blok kendali *on-off.* Daerah dengan sinyal pembangkit kesalahan yang digerakkan sebelum terjadi *switching* disebut *dead band. Dead band* ditunjukkan pada Gambar 2.23. *Dead band* ini menyebabkan keluaran kendali u(t) tetap pada nilai awal sampai sinyal pembangkit kesalahan bergerak mendekati nilai nol. Dalam

beberapa hal *dead band* terjadi sebagai akibat adanya penghalang yang tidak diinginkan dan gerakan yang hilang, sering juga hal ini dimaksudkan untuk mencegah operasi yang berulang-ulang dari mekanisme *on-off*.