

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Sensor dan Transduser**

D Sharon, dkk (1982), mengatakan sensor adalah suatu peralatan yang berfungsi untuk mendeteksi gejala-gejala atau sinyal-sinyal yang berasal dari perubahan suatu energi seperti energi listrik, energi fisika, energi kimia, energi biologi, energi mekanik dan sebagainya..

**Contoh;** Camera sebagai sensor penglihatan, telinga sebagai sensor pendengaran, kulit sebagai sensor peraba, LDR (*light dependent resistance*) sebagai sensor cahaya, dan lainnya.

William D.C, (1993), mengatakan transduser adalah sebuah alat yang bila digerakan oleh suatu energi di dalam sebuah sistem transmisi, akan menyalurkan energi tersebut dalam bentuk yang sama atau dalam bentuk yang berlainan ke sistem transmisi berikutnya". Transmisi energi ini bisa berupa listrik, mekanik, kimia, optic (radiasi) atau thermal (panas).

**Contoh:** generator adalah transduser yang merubah energi mekanik menjadi energi listrik, motor adalah transduser yang merubah energi listrik menjadi energi mekanik, dan sebagainya.

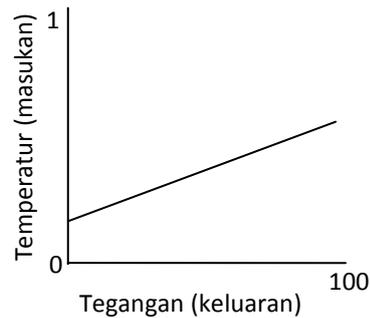
##### **2.1.1 Peryaratan Umum Sensor dan Transduser**

Dalam memilih peralatan sensor dan transduser yang tepat dan sesuai dengan sistem yang akan disensor maka perlu diperhatikan persyaratan umum sensor berikut ini : (D Sharon, dkk, 1982)

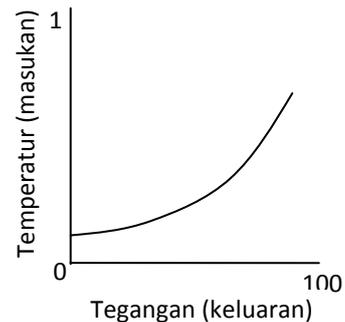
###### **a. Linearitas**

Ada banyak sensor yang menghasilkan sinyal keluaran yang berubah secara kontinyu sebagai tanggapan terhadap masukan yang berubah secara kontinyu. Sebagai contoh, sebuah sensor panas dapat menghasilkan tegangan sesuai dengan panas yang dirasakannya. Dalam kasus seperti ini, biasanya dapat diketahui secara tepat bagaimana perubahan keluaran dibandingkan dengan

masukannya berupa sebuah grafik. **Gambar 2.1** memperlihatkan hubungan dari dua buah sensor panas yang berbeda. Garis lurus pada **gambar 2.1(a)**, memperlihatkan tanggapan linier, sedangkan pada **gambar 2.1(b)** adalah tanggapan non-linier.



(a) Tanggapan linier



(b) Tanggapan non linier

**Gambar 2.1** Keluaran dari transduser panas

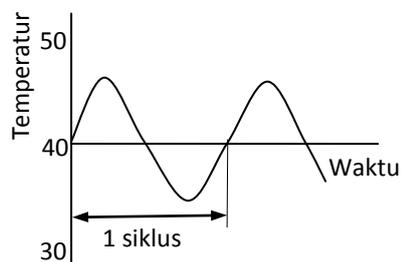
#### b. Sensitivitas

Sensitivitas akan menunjukkan seberapa jauh kepekaan sensor terhadap kuantitas yang diukur. Sensitivitas sering juga dinyatakan dengan bilangan yang menunjukkan “perubahan keluaran dibandingkan unit perubahan masukan”. Beberapa sensor panas dapat memiliki kepekaan yang dinyatakan dengan “satu volt per derajat”, yang berarti perubahan satu derajat pada masukan akan menghasilkan perubahan satu volt pada keluarannya. Sensor panas lainnya dapat saja memiliki kepekaan “dua volt per derajat”, yang berarti memiliki kepekaan dua kali dari sensor yang pertama. Linieritas sensor juga mempengaruhi sensitivitas dari sensor. Apabila tanggapannya linier, maka sensitivitasnya juga akan sama untuk jangkauan pengukuran keseluruhan. Sensor dengan tanggapan pada **gambar 2.1(b)** akan lebih peka pada temperatur yang tinggi dari pada temperatur yang rendah.

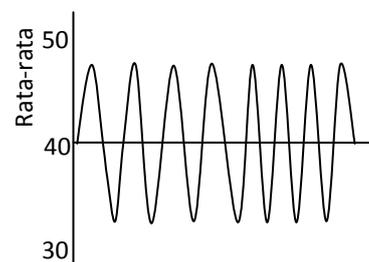
### c. Tanggapan Waktu

Tanggapan waktu pada sensor menunjukkan seberapa cepat tanggapannya terhadap perubahan masukan. Sebagai contoh, instrumen dengan tanggapan frekuensi yang jelek adalah sebuah termometer merkuri. Masukannya adalah temperatur dan keluarannya adalah posisi merkuri. Misalkan perubahan temperatur terjadi sedikit demi sedikit dan kontinyu terhadap waktu, seperti tampak pada **gambar 2.2(a)**.

Frekuensi adalah jumlah siklus dalam satu detik dan diberikan dalam satuan hertz (Hz). [1 hertz berarti 1 siklus per detik, 1 kilohertz berarti 1000 siklus per detik]. Pada frekuensi rendah, yaitu pada saat temperatur berubah secara lambat, termometer akan mengikuti perubahan tersebut dengan “setia”. Tetapi apabila perubahan temperatur sangat cepat lihat **gambar 2.2(b)** maka tidak diharapkan akan melihat perubahan besar pada termometer merkuri, karena ia bersifat lamban dan hanya akan menunjukkan temperatur rata-rata.



(a) Perubahan lambat



(b) Perubahan cepat

### Gambar 2.2 Temperatur berubah secara kontinyu

Ada bermacam cara untuk menyatakan tanggapan frekuensi sebuah sensor. Misalnya “satu milivolt pada 500 hertz”. Tanggapan frekuensi dapat pula dinyatakan dengan “*decibel (db)*”, yaitu untuk membandingkan daya keluaran pada frekuensi tertentu dengan daya keluaran pada frekuensi referensi.

Yayan I.B, (1998), mengatakan ketentuan lain yang perlu diperhatikan dalam memilih sensor yang tepat adalah dengan mengajukan beberapa pertanyaan berikut ini:



- a. Apakah ukuran fisik sensor cukup memenuhi untuk dipasang pada tempat yang diperlukan?
- b. Apakah ia cukup akurat?
- c. Apakah ia bekerja pada jangkauan yang sesuai?
- d. Apakah ia akan mempengaruhi kuantitas yang sedang diukur?.

Sebagai contoh, bila sebuah sensor panas yang besar dicelupkan kedalam jumlah air yang kecil, malah menimbulkan efek memanaskan air tersebut, bukan menyensornya.

- e. Apakah ia tidak mudah rusak dalam pemakaiannya?.
- f. Apakah ia dapat menyesuaikan diri dengan lingkungannya?
- g. Apakah biayanya terlalu mahal?

### 2.1.2 Klasifikasi Sensor

Secara umum berdasarkan fungsi dan penggunaannya sensor dapat dikelompokkan menjadi 3 bagian yaitu:

- a. sensor thermal (panas)
- b. sensor mekanis
- c. sensor optik (cahaya)

Sensor thermal adalah sensor yang digunakan untuk mendeteksi gejala perubahan panas/temperature/suhu pada suatu dimensi benda atau dimensi ruang tertentu.

**Contohnya:** *bimetal, termistor, termokopel, RTD, photo transistor, photo dioda, photo multiplier, photovoltaik, infrared pyrometer, hygrometer, dsb.*

Sensor mekanis adalah sensor yang mendeteksi perubahan gerak mekanis, seperti perpindahan atau pergeseran atau posisi, gerak lurus dan melingkar, tekanan, aliran, level dsb.

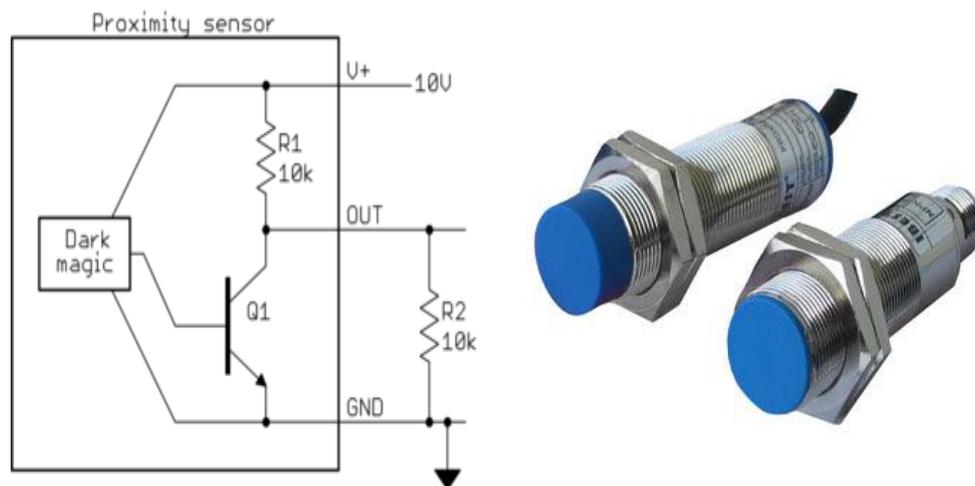
**Contoh:** *strain gage, linear variable deferential transformer (LVDT), proximity, potensiometer, load cell, bourdon tube, dsb.*

Sensor optic atau cahaya adalah sensor yang mendeteksi perubahan cahaya dari sumber cahaya, pantulan cahaya ataupun bias cahaya yang mengenai benda atau ruangan.

**Contoh;** *photo cell, photo transistor, photo diode, photo voltaic, photo multiplier, pyrometer optic, dsb.*

### 2.1.3 Sensor Proximity

Proximity Switch atau Sensor Proximity adalah alat pendeteksi yang bekerja berdasarkan jarak obyek terhadap sensor. Karakteristik dari sensor ini adalah mendeteksi obyek benda dengan jarak yang cukup dekat, berkisar antara 1 mm sampai beberapa centi meter saja sesuai type sensor yang digunakan. Proximity Switch ini mempunyai tegangan kerja antara 10-30 Vdc dan ada juga yang menggunakan tegangan 100-200VAC. **Gambar 2.3** adalah merupakan wiring diagram dan bentuk fisik dari sensor proximity secara umum seperti di bawah ini.



**Gambar 2.3** Wiring Diagram dan Bentuk fisik Sensor Proximity

#### 2.1.3.1 Prinsip kerja Proximity Switch

Sensor ini bekerja berdasarkan jarak object terhadap sensor, Pada sensor proximity ini dialiri tegangan sebesar 24 VDC. Prinsip kerja sensor proximity yaitu sama seperti saklar. Posisi normal pada sensor ini yaitu *normally open*, ketika ada objek logam yang mendekat kepada sensor dengan jarak yang sangat

dekat 5 mm misalkan, maka pada sensor tersebut akan terhubung dari kontak *normally open* menjadi *normally close* yang terhubung ke perangkat PLC dan Mikrokontroler. Kabel-kabel yang tersedia bisa dihubungkan ke perangkat lainnya seperti PLC, lampu indikator, relay dll. Pada saat sensor ini sedang bekerja atau mendeteksi adanya logam (besi) maka akan ditandai dengan lampu kecil berwarna merah atau hijau yang ada dibagian atas sensor, sehingga memudahkan kita dalam memonitor kerja sensor atau ketika melakukan *preventive maintenance*.

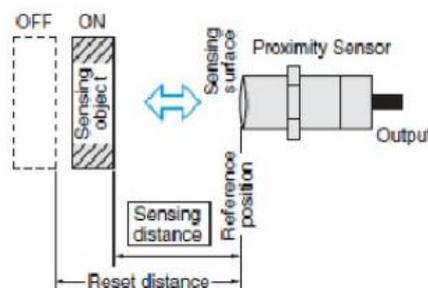
Hampir setiap mesin - mesin produksi yang ada di setiap industri baik itu industri kecil ataupun besar menggunakan sensor jenis ini sebab selain praktis sensor ini termasuk tahan terhadap benturan ataupun guncangan selain itu mudah pada saat melakukan perawatan ataupun penggantian, sebab telah dirancang demikian oleh produsennya. Adapun salah satu contoh penggunaan atau penerapan dari sensor jenis ini adalah digunakan untuk mendeteksi gerakan *cylinder up* atau *down* pada sebuah mesin atau penggerak.

#### Contoh Penerapan :



**Gambar 2.4 Prinsip kerja sensor proximity**

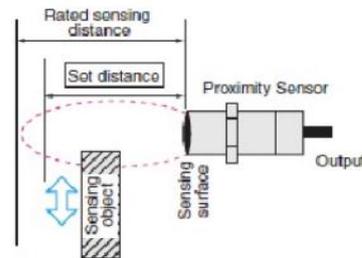
(Sumber: <http://id.scribd.com/doc/116434840/Prinsip-Kerja-Proximity-Sensor#scribd>)



**Gambar 2.5 Sensor Proximity sebelum pengukuran**

(Sumber: <http://id.scribd.com/doc/116434840/Prinsip-Kerja-Proximity-Sensor#scribd>)

Jarak dari permukaan referensi yang memungkinkan penggunaan stabil, termasuk pengaruh suhu dan tegangan, ke posisi objek (standar) sensing transit. Ini adalah sekitar 70% sampai 80% dari jarak (nilai) normal sensing.

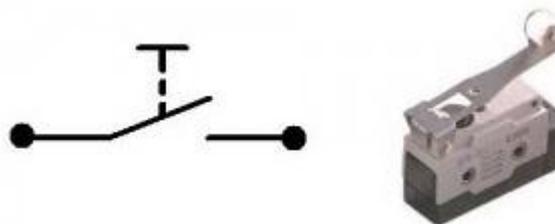


**Gambar 2.6** Jarak deteksi sensor proximity

(Sumber: <http://id.scribd.com/doc/116434840/Prinsip-Kerja-Proximity-Sensor#scribd>)

## 2.2 *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 menghubungkan pada saat katupnya ditekan pada batas penekanan tertentu yang telah ditentukan dan akan memutus 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* ditunjukkan pada gambar berikut.



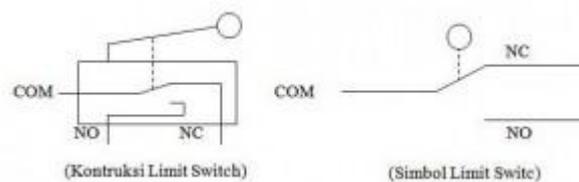
**Gambar 2.7** Simbol Dan Bentuk *Limit Switch*

(Sumber :<http://elektronika-dasar.web.id/komponen/limit-switch-dan-saklar-push-on/>)

*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 dan simbol limit switch dapat dilihat seperti gambar di bawah.



**Gambar 2.8 Konstruksi Dan Simbol *Limit Switch***

(Sumber :<http://elektronika-dasar.web.id/komponen/limit-switch-dan-saklar-push-on/>)

### 2.3 *Push Button*

Saklar merupakan komponen elektronika yang berfungsi untuk menghubungkan dan memutuskan dua titik atau lebih dalam suatu rangkaian elektronika. Salah satu jenis saklar adalah saklar *Push button* yaitu saklar yang hanya akan menghubungkan dua titik atau lebih pada saat tombolnya ditekan dan pada saat tombolnya tidak ditekan maka akan memutuskan dua titik atau lebih dalam suatu rangkaian elektronika. Bentuk saklar *Push button* ditunjukkan pada gambar berikut.



**Gambar 2.9 Saklar *Push Button***

(Sumber: <http://elektronika-dasar.web.id/komponen/limit-switch-dan-saklar-push-on/>)

Saklar *push button* dapat berbentuk berbagai macam, ada yang menggunakan tuas dan ada yang tanpa tuas. Saklar *push button* sering diaplikasikan pada tombol-tombol perangkat elektronik digital. Salah satu contoh penggunaan saklar *push ON* adalah pada keyboard komputer, keypad printer, matrik keypad, tombol kontrol pada DVD player dan lain sebagainya.

#### **2.4 Pengenalan Mikrokontroler ATMEGA 8535**

Mikrokontroler merupakan suatu komponen elektronika yang didalamnya terdapat rangkaian mikroprosesor, memori (RAM/ROM) dan I/O, rangkaian tersebut terdapat dalam *level chip* atau biasa disebut *single chip microcomputer*. Pada Mikrokontroler sudah terdapat komponen-komponen mikroprosesor dengan bus-bus internal yang saling berhubungan. Komponen-komponen tersebut adalah RAM, ROM, *timer*, komponen I/O paralel dan serial, dan *interrupt controller*. Adapun keunggulan dari Mikrokontroler adalah adanya sistem *interrupt*. Sebagai perangkat kontrol penyesuaian, mikrokontroler sering disebut juga untuk menaikkan respon eksternal (*interrupt*) pada waktu yang nyata. Perangkat tersebut harus melakukan hubungan *switching* cepat, menunda satu proses ketika adanya respon eksekusi yang lain.

AVR merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur RISC (*Reduced Instruction Set Computer*). Hampir semua instruksi dieksekusi dalam satu siklus clock. AVR mempunyai 32 *register general-purpose*, *timer/counter* fleksibel dengan mode *compare*, *interrupt* internal dan eksternal, serial

UART, *programmable Watchdog Timer*, dan *mode power saving*. Beberapa diantaranya mempunyai ADC dan PWM internal. AVR juga mempunyai *In- System Programmable Flash on-chip* yang mengizinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI (*Serial Peripheral Inteface*). Chip AVR yang digunakan untuk tugas akhir ini adalah jenis IC Mikrokontroler ATMEGA 8535. ATMEGA 8535 adalah mikrokontroler CMOS 8-bit daya-rendah berbasis arsitektur RISC yang ditingkatkan. Kebanyakan instruksi dikerjakan pada satu siklus *clock*, ATmega 8535 mempunyai *throughput* mendekati 1 MIPS per MHz membuat desain sistem untuk mengoptimalkan konsumsi daya dengan kecepatan proses. AVR ATmega8535 mempunyai 40 kaki, 32 kaki yang digunakan untuk keperluan port paralel setiap port terdiri dari 8 pin, sehingga terdapat 4 port, yaitu port A, port B, port C, port D. seperti gambar dibawah ini :

(XCK/T0) PB0	1	40	PA0 (ADC0)
(T1) PB1	2	39	PA1 (ADC1)
(INT2/AIN0) PB2	3	38	PA2 (ADC2)
(OC0/AIN1) PB3	4	37	PA3 (ADC3)
(SS) PB4	5	36	PA4 (ADC4)
(MOSI) PB5	6	35	PA5 (ADC5)
(MISO) PB6	7	34	PA6 (ADC6)
(SCK) PB7	8	33	PA7 (ADC7)
RESET	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL2	12	29	PC7 (TOSC2)
XTAL1	13	28	PC6 (TOSC1)
(RXD) PD0	14	27	PC5
(TXD) PD1	15	26	PC4
(INT0) PD2	16	25	PC3
(INT1) PD3	17	24	PC2
(OC1B) PD4	18	23	PC1 (SDA)
(OC1A) PD5	19	22	PC0 (SCL)
(ICP1) PD6	20	21	PD7 (OC2)

**Gambar 2.10 Konfigurasi Pin AVR ATMEGA 8535**

(Sumber : <http://elib.unikom.ac.id/download.php?id=192704>)

Berikut adalah penjelasan fungsi tiap kaki yang biasa ada pada seri mikrokontroler AVR ATmega8535 yaitu :

1. **VCC (power supply)**
2. **GND (ground)**
3. **Port A (PA0-PA7)**

Port A berfungsi sebagai *input* analog pada A/D Konverter. Port A juga berfungsi sebagai suatu Port I/O 8-bit dua arah, jika A/D Konverter tidak digunakan. Pin - pin Port dapat menyediakan resistor *internal pull-up* (yang dipilih



---

untuk masing-masing bit). Port A *output buffer* mempunyai karakteristik gerakan simetris. Ketika pin PA0 ke PA7 digunakan sebagai *input* dan secara eksternal ditarik rendah, pin-pin akan memungkinkan arus sumber jika resistor *internal pull-up* diaktifkan. Pin Port A adalah *tri-stated* manakala suatu kondisi *reset* menjadi aktif, sekalipun waktu habis.

**1. Port B (PB0-PB7)**

Port B adalah suatu Port I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). Port B *output buffer* mempunyai karakteristik gerakan simetris. Sebagai *input*, pin port B yang secara eksternal ditarik rendah akan arus sumber jika resistor *pullup* diaktifkan. Pin Port B adalah *tri-stated* pada suatu kondisi tertentu.

**2. Port C (PC0-PC7)**

Port C adalah suatu Port I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). Port C *output buffer* mempunyai karakteristik gerakan simetris. Sebagai *input*, pin port C yang secara eksternal ditarik rendah akan arus sumber jika resistor *pullup* diaktifkan. Pin Port C adalah *tri-stated* pada suatu kondisi tertentu.

**3. Port D (PD0-PD7)**

Port D adalah suatu Port I/O 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). Port D *output buffer* mempunyai karakteristik gerakan simetris. Sebagai *input*, pin port D yang secara eksternal ditarik rendah akan arus sumber jika resistor *pullup* diaktifkan. Pin Port D adalah *tri-stated* pada suatu kondisi tertentu.

**7. RESET (Reset input)**

**8. XTAL1 (Input Oscillator)**

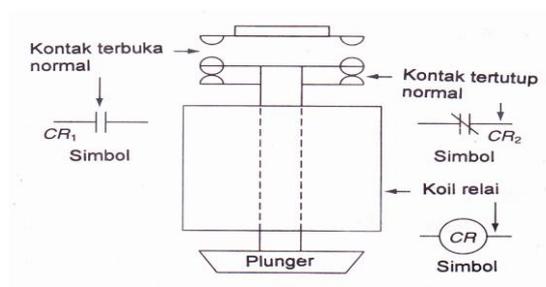
**9. XTAL2 (Output Oscillator)** AVCC adalah pin penyedia tegangan untuk port A dan A/D Konverter.

**10. AREF** adalah pin referensi analog untuk A/D konverter.

---



2. *Normally Close (NC)*, bila diberi tegangan yang mencukupi pada kumparannya maka kontak penghubung menjadi terbuka.
3. *Change Over (CO)*, *relay* ini mempunyai saklar tunggal yang normalnya tertutup yang lama, bila kumparan 1 dialiri arus maka saklar akan terhubung ke terminal A, sebaliknya bila kumparan 2 dialiri arus maka saklar akan terhubung ke terminal B.



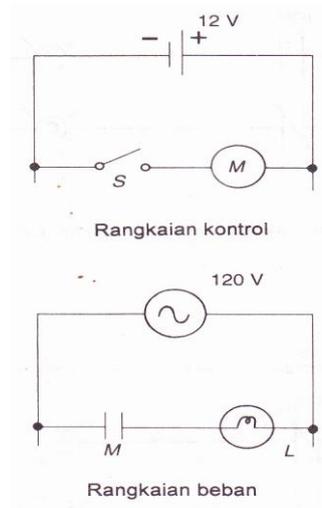
**Gambar 2.12 Relay elektromagnetis (*electromechanical relay* = EMR)**

(Sumber: PETRUZELLA, FRANK D. 2001. “Elektronik Industri”.Edisi Kedua. Penerbit Andi Yogyakarta.)

Kontak *normally open* akan membuka ketika tidak ada arus mengalir pada kumparan, tetapi tertutup secepatnya setelah kumparan menghantarkan arus atau diberi tenaga. Kontak *normally close* akan tertutup apabila kumparan tidak diberi daya dan membuka ketika kumparan diberi daya. Masing-masing kontak biasanya digambarkan sebagai kontak yang tampak dengan kumparan tidak diberi daya. Sebagian besar *relay* kontrol mesin mempunyai beberapa ketentuan untuk pengubahan kontak *normally open* menjadi *normally closed* atau sebaliknya. Itu berkisar dari kontak sederhana “*flip-over*” untuk melepaskan kontak dan menempatkan kembali dengan perubahan lokasi pegas.

Banyak EMR yang mempunyai beberapa perangkat kontak yang dioperasikan dengan kumparan tunggal. Misalnya *relay* yang digunakan untuk mengontrol beberapa operasi penghubungan dengan arus tunggal terpisah. Pada umumnya *relay* kontrol digunakan sebagai alat pembantu untuk kontrol penghubungan rangkaian dan beban. Misalnya, motor kecil, solenoida dan lampu pilot. EMR dapat digunakan untuk mengontrol rangkaian beban tegangan tinggi dengan rangkaian kontrol tegangan rendah. Ini memungkinkan sebab kumparan dan kontak dari *relay* secara listrik terisolasi satu sama lain. Dari segi keamanan,

rangkaian tersebut mempunyai perlindungan ekstra bagi operator. Misalnya, anggap bahwa anda ingin menggunakan *relay* untuk mengontrol rangkaian lampu 120 V dengan rangkaian kontrol 12 V. Lampu akan dirangkai seri dengan kontak *relay* pada sumber 120 V (**gambar 2.13**). Saklar akan dirangkai seri terhadap kumparan *relay* pada sumber 12 V. Pengoperasian saklar adalah dengan memberi energi atau menghilangkan energi kumparan. Hal ini pada gilirannya akan menutup atau membuka kontak pada saklar *ON* atau *OFF* lampu.

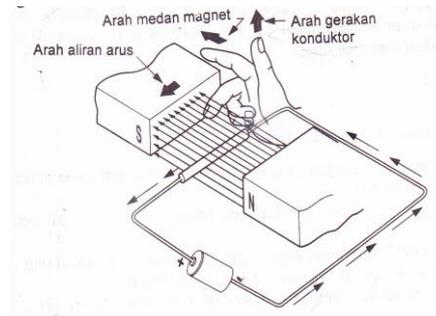


**Gambar 2.13 Penggunaan *relay* untuk mengontrol rangkaian beban tegangan tinggi dengan rangkaian kontrol tegangan rendah**

(Sumber: PETRUZELLA, FRANK D. 2001. “Elektronik Industri”.Edisi Kedua. Penerbit Andi Yogyakarta.)

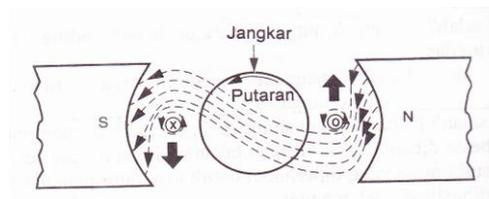
## 2.6 Motor DC

Untuk menentukan arah gerakan penghantar yang mengalirkan arus pada medan magnet, digunakan hukum tangan kanan motor (**gambar 2.14 (a)**). Ibu jari dan dua jari yang pertama dari tangan kanan disusun sehingga saling tegak lurus satu sama lain dengan menunjukkan arah garis gaya magnet dari medan, dan jari tengah menunjukkan arah arus yang mengalir (*min* ke *plus*) pada penghantar. Ibu jari akan menunjukkan arah gerakan penghantar, seperti diperlihatkan pada (**gambar 2.14 (b)**)



**Gambar 2.14 (a) Hukum tangan kanan motor**

(Sumber: PETRUZELLA, FRANK D. 2001. "Elektronik Industri".Edisi Kedua. Penerbit Andi Yogyakarta.)

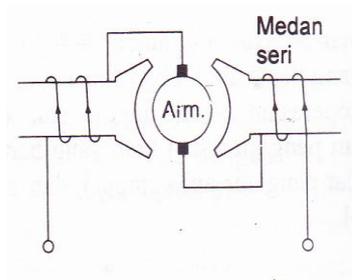


**Gambar 2.14 (b) Prinsip motor**

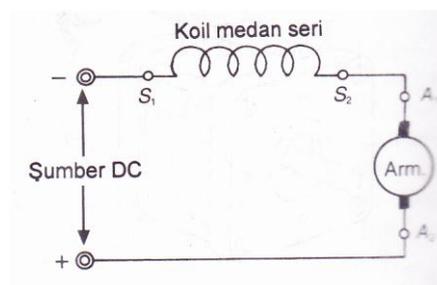
(Sumber: PETRUZELLA, FRANK D. 2001. "Elektronik Industri".Edisi Kedua. Penerbit Andi Yogyakarta.)

Motor DC jenis seri (**gambar 2.15**) terdiri dari medan seri (diidentifikasi dengan  $S_1$  dan  $S_2$ ) dibuat dari sedikit lilitan kawat besar yang dihubungkan seri dengan jangkar (diidentifikasi dengan  $S_1$  dan  $S_2$ ). Jenis motor DC ini mempunyai karakteristik torsi start dan kecepatan variabel yang tinggi. Ini berarti bahwa motor dapat start atau menggerakkan beban yang sangat berat, tetapi kecepatan akan bertambah kalau beban turun. Motor Dc seri dapat membangkitkan torsi *starting* yang besar karena arus yang sama yang melewati jangkar juga melewati medan. Jadi, jika jangkar memerlukan arus yang banyak (membangkitkan torsi yang besar), arus ini juga melewati medan, menambah kekuatan medan. Oleh karena itu, motor seri berputar cepat dengan beban ringan dan berputar lambat pada saat beban ditambahkan.

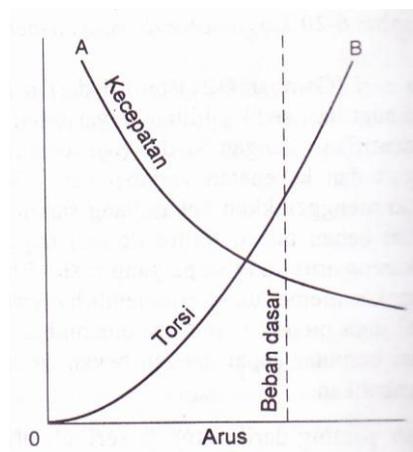
Sifat istimewa terpenting motor DC seri adalah kemampuannya untuk start atau menjalankan beban yang sangat berat. Karena alasan itu, motor sering digunakan pada gerakan dan elevator. Untuk membalik arah putaran motor DC seri, balik arah arus pada kumparan seri atau kumparan jangkar.



(a) Diagram pengawatan



(b) Diagram skematik



(c) Karakteristik kecepatan dan torsi

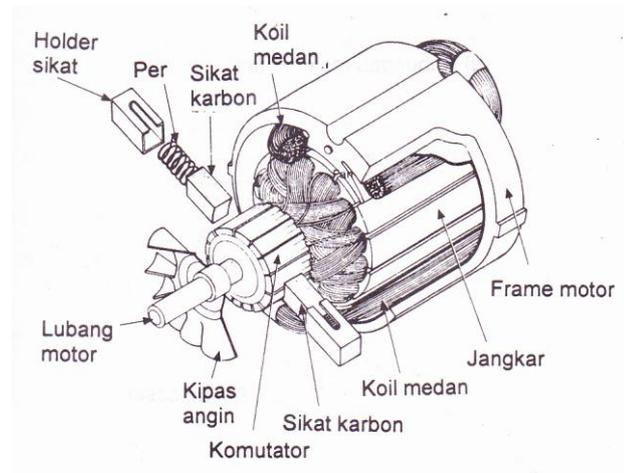
### Gambar 2.15 Motor DC Seri

(Sumber: PETRUZELLA, FRANK D. 2001. "Elektronik Industri". Edisi Kedua. Penerbit Andi Yogyakarta.)

Jenis motor DC ini juga sering disebut motor *universal* karena dapat dioperasikan baik dengan arus searah maupun dengan arus bolak-balik (**gambar**

2.15). Alasan untuk ini adalah bahwa motor DC akan terus pada arah yang sama jika arus yang mengalir pada jangkar dan arus yang mengalir pada medan dibalik pada waktu bersamaan.

Penggerak sabuk (*belt*) tidak pernah digunakan pada motor seri. Motor tersebut sering dirangkai langsung dengan beban karena motor seri tidak dapat dioperasikan dalam keadaan tanpa beban. Fluks magnet yang ada pada keadaan tanpa beban disebabkan oleh magnet sisa, karena itu medan tersebut sangat lemah. Pengoperasian motor tanpa beban akan memungkinkan rotor mencapai kecepatan yang sedemikian tinggi sehingga gaya sentrifugal akan menyebabkan lilitan menjadi rusak. Pada motor DC seri kecil gesekan sikat, gesekan bantalan dan kerugian kumparan dapat menimbulkan beban yang cukup untuk menahan turunnya kecepatan sampai level yang aman. Ketika digunakan pada alat daya yang mudah dibawa, “train gear” pada unti dan alat yang dipasangkan akan memberikan beban yang cukup.



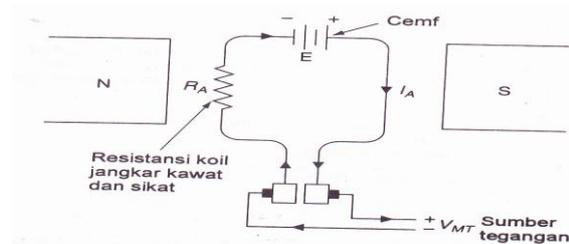
**Gambar 2.16 Bagian bagian motor DC seri**

(Sumber: PETRUZELLA, FRANK D. 2001. “Elektronik Industri”.Edisi Kedua. Penerbit Andi Yogyakarta.)

Biasanya motor dipasang untuk mengerjakan pekerjaan tertentu yang memerlukan arah putaran yang tepat, tetapi mungkin pada suatu waktu anda ingin meruba arah putaran. Arah putaran motor Dc tergantung pada arah medan dan arah aliran arus pada jangkar. Jika arah medan atau arah aliran arus pada jangkar

dibalik, putaran motor akan terbalik. Jika kedua faktor tersebut dibalik pada saat yang sama, motor akan terus berputar pada arah putaran yang sama. Untuk membalik arah putaran motor compound baliklah ujung-ujung jangkar atau pada medan seri dan shunt. Standar industri adalah dengan membalik arus pada jangkar. Motor DC seperti pada generator DC mempunyai reaksi jangkar. Karena jangkar motor mempunyai arus yang mengalir padanya, medan magnet akan dibangkitkan disekitar kumparan jangkar sebagai akibat dari arus tersebut. Medan jangkar menyebabkan distorsi medan utama pada motor, yang menyebabkan garis netral tergeser.

Arah distorsi pada motor berlawanan dengan arah distorsi pada generator. Pada motor, reaksi jangkar menggeser garis netral berlawanan dengan arah putaran. Kutub bantu (*interpole*) digunakan sama pada motor DC untuk mencegah bunga api pada sikat-sikat. Pada saat jangkar berputar, kumparan jangkar DC memotong medan magnet dan menginduksikan tegangan atau GGL pada kumparan tersebut. Hal ini kadang-kadang disebut aksi generator pada motor. Karena tegangan yang diinduksikan bertentangan dengan tegangan terminal yang diberikan, tegangan ini disebut GGL lawan (*counter electromotive force = cemf*). Tegangan efektif yang bekerja pada rangkaian jangkar daro motor yang diberikan, atau tegangan terminal diberikan GGL lawan (**gambar 2.17**).



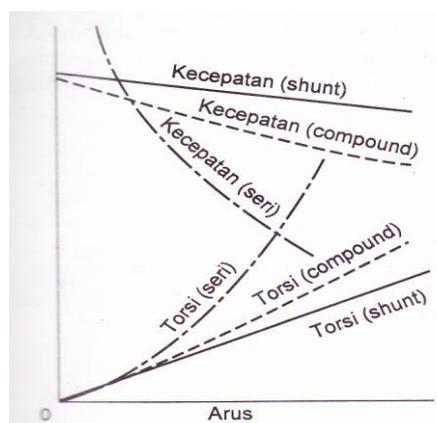
**Gambar 2.17 GGL lawan motor**

(Sumber: PETRUZELLA, FRANK D. 2001. "Elektronik Industri". Edisi Kedua. Penerbit Andi Yogyakarta.)

Kekuatan GGL lawan pada motor ditentukan dengan jumlah penghantar jangkar yang diseri antara sikat-sikat dan kecepatan jangkar. Pada saat motor *start*, jangkar tidak berputar sehingga tidak ada GGL lawan. Akibatnya tegangan lin penuh diberikan pada motor dan akan menarik arus pada rangkaian jangkar

sesuai dengan hukum Ohm. Salah satu faktor yang membatasi arus adalah tahanan lilitan motor. Ketika kecepatan jangkar bertambah atau bertambah cepat, ketika GGL lawan akan timbul dan membatasi aliran arus pada motor. Kecepatan bertambah, GGL lawan bertambah dan arus yang ditarik pada motor berkurang. Ketika motor mencapai kecepatan penuh pada keadaan tanpa beban, akan ada pembangkitan GGL yang hampir sama pada tegangan lin. Hanya arus yang cukup yang dapat mengalir dan mempertahankan kecepatan ini. Jika diberikan beban pada motor, maka kecepatan berkurang dan mengurangi GGL lawan, sehingga arus yang lebih banyak akan ditarik untuk menggunakan beban. Output beban motor mengatur kecepatan dengan mempengaruhi GGL lawan dan aliran arus.

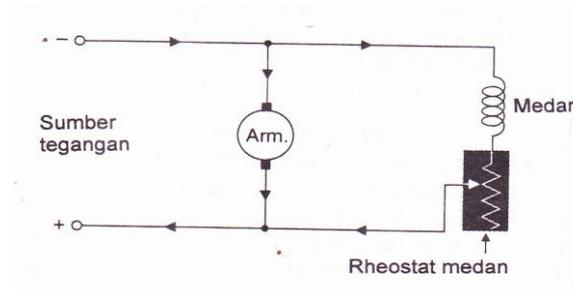
Kecepatan motor DC tergantung pada kekuatan medan magnet dan tegangan yang diberikan pada jangkar dan juga pada beban. Oleh karena itu, kecepatan dapat diatur baik dengan mengatur arus medan atau dengan mengatur tegangan yang diberikan pada jangkar. Apabila beban bertambah, maka kecepatan dan GGL lawan menurun dan arus bertambah. Demikian juga apabila beban menurun, kecepatan dan GGL lawan bertambah dan arus p. Motor dirancang untuk menghasilkan *horse power* (hp) kerja pada kecepatan penuh. Kecepatan normalnya (beban penuh) disebut kecepatan basis pada motor. Kecepatan basis ini diperoleh ketika tegangan kerja jangkar dan arus beban penuh diberikan (**gambar 2.18**).



**Gambar 2.18** Pengaturan kecepatan motor DC

(Sumber: PETRUZELLA, FRANK D. 2001. "Elektronik Industri". Edisi Kedua. Penerbit Andi Yogyakarta.)

Kecepatan motor DC sebanding dengan GGL lawan. Pelemahan fluks medan utama mengurangi GGL lawan. Semakin rendah GGL lawan, semakin banyak arus yang mengalir pada rangkaian jangkar. Penambahan arus jangkar ini menyebabkan medan magnet yang lebih kuat pada jangkar sehingga kecepatan jangkar bertambah. Kecepatan bertambah sampai GGL lawan dapat membatasi arus jangkar pada harga yang baru. Harga tersebut ditentukan oleh kekuatan medan utama. Pada titik tersebut motor menggerakkan beban pada kecepatan konstan. Pada motor shunt pengaturan kecepatan dapat dilakukan dengan menghubungkan rheostat seri dengan lilitan medan shunt. Penambahan tahanan seri pada medan mengurangi arus medan sehingga memperlemah medan magnet. Penurunan kekuatan medan berarti motor harus berputar lebih cepat (**gambar 2.19**). Metode pengaturan motor ini sering digunakan karena sederhana dan murah. Kecepatan yang lebih besar daripada kecepatan yang basis hanya ada pada torsi yang dikurangi. Pelemahan medan mengurangi kemampuan torsi motor.



**Gambar 2.19 Pengaturan kecepatan medan**

(Sumber: PETRUZELLA, FRANK D. 2001. "Elektronik Industri". Edisi Kedua. Penerbit Andi Yogyakarta.)

Penggunaan motor DC akhir-akhir ini mengalami perkembangan, khususnya dalam pemakaiannya sebagai motor penggerak. Motor DC digunakan secara luas pada berbagai motor penggerak dan pengangkut dengan kecepatan yang bervariasi yang membutuhkan respon dinamis dan keadaan steady-state. Motor DC mempunyai pengaturan yang sangat mudah dilakukan dalam berbagai kecepatan dan beban yang bervariasi. Itu sebabnya motor DC digunakan pada berbagai aplikasi tersebut contohnya pada sistem kendali prototipe pintu otomatis pada lift 3 lantai.



**Gambar 2.20 Motor DC Penggerak Pintu Otomatis**

Sumber : Dokumentasi penulis

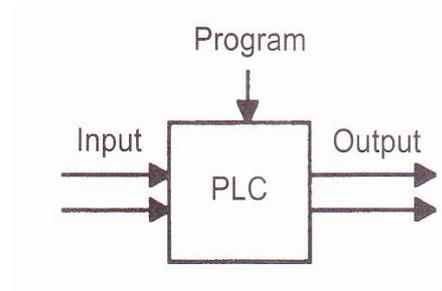
## 2.7 Pintu Otomatis

Perhatikan sebuah pintu otomatis yang akan membuka apabila seseorang mendekatinya, tetap terbuka selama jangka waktu yang telah ditetapkan, misalnya 10 detik, sebelum akhirnya menutup kembali. Input-input ke sistem kontrol dapat diberikan oleh sebuah sensor yang mendeteksi seseorang yang mendekati pintu dari luar dan sebuah sensor lainnya yang mendeteksi seseorang yang mendekati pintu dari dalam. Sensor-sensor ini dapat berupa elemen-elemen semikonduktor yang sensitif terhadap panas, yang akan memberikan sinyal-sinyal tegangan ketika radiasi inframerah menerpanya. Terdapat juga input-input ke pengontrol yang mungkin diberikan oleh saklar-saklar limit untuk mengindikasikan saat di mana pintu terbuka sepenuhnya dan oleh sebuah timer untuk mempertahankan terbukanya pintu selama jangka waktu yang ditentukan. Output dari pengontrol dapat diberikan pada katup-katup pneumatik yang dioperasikan oleh solenoida, yang memanfaatkan gerakan piston-piston di dalam silinder untuk membuka dan menutup pintu.

## 2.8 PLC (*Programmable Logic Controller*)

*Programmable logic controller* singkatnya PLC merupakan suatu bentuk khusus pengontrol berbasis-mikroprosesor yang memanfaatkan memori yang dapat diprogram untuk menyimpan instruksi-instruksi dan untuk mengimplementasikan fungsi-fungsi semisal logika, *sequencing*, pewaktuan (*timing*), pencacahan (*counting*) dan aritmatika guna mengontrol mesin-mesin dan proses-proses (**Gambar 2.21**) dan dirancang untuk dioperasikan oleh para

insinyur yang hanya memiliki sedikit pengetahuan mengenai komputer dan bahasa pemrograman.



**Gambar 2.21 Sebuah PLC**

(Sumber: Bolton, William. 2004. “Programmable Logic Controller (PLC) Sebuah Pengantar”. Edisi Ketiga. Penerbit Erlangga)

Piranti ini dirancang sedemikian rupa agar tidak hanya para programmer komputer saja yang dapat membuat atau mengubah program-programnya. Oleh karena itu, para perancang PLC telah menempatkan sebuah program awal di dalam piranti ini (*pre-program*) yang memungkinkan program-program kontrol dimasukkan dengan menggunakan suatu bentuk bahasa pemrograman yang sederhana dan intuitif. Istilah logika (*logic*) dipergunakan karena pemrograman yang harus dilakukan sebagian besar berkaitan dengan pengimplementasian operasi-operasi logika dan penyambungan (*switching*), misalnya jika A atau B terjadi maka sambungkan (atau hidupkan) C, jika A dan B terjadi maka sambungkan D. Perangkat-perangkat input yaitu sensor-sensor semisal saklar, dan perangkat-perangkat output di dalam sistem dikontrol, misalnya motor, katup dsb. Disambungkan ke PLC sang operator kemudian memasukkan serangkaian instruksi yaitu sebuah program ke dalam memori PLC. Perangkat pengontrol tersebut kemudian memantau input-input dan output-output sesuai dengan instruksi-instruksi di dalam program dan melaksanakan aturan-aturan kontrol yang telah diprogramkan.

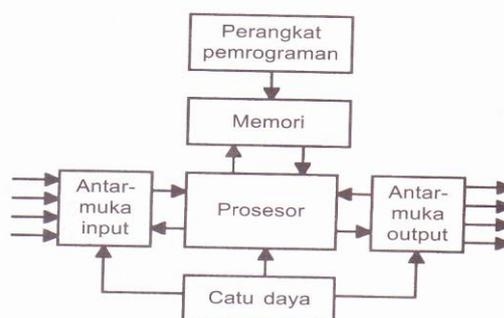
PLC memiliki keunggulan yang signifikan, karena sebuah perangkat pengontrol yang sama dapat dipergunakan di dalam beraneka ragam sistem kontrol. Untuk memodifikasi sebuah sistem kontrol dan aturan-aturan

pengontrolan yang dijalankannya, yang harus dilakukan oleh seorang operator hanyalah melakukan seperangkat instruksi yang berbeda dari yang digunakan sebelumnya. Penggantian rangkaian kontrol tidak perlu dilakukan. Hasilnya adalah sebuah perangkat yang fleksibel dan hemat biaya yang dapat dipergunakan dalam sistem-sistem kontrol yang sifat dan kompleksitasnya sangat beragam.

PLC serupa dengan komputer namun, bedanya: komputer dioptimalkan untuk tugas-tugas penghitung dan penyajian data, sedangkan PLC dioptimalkan untuk tugas-tugas pengontrolan dan pengoperasian di dalam lingkungan industri. Dengan demikian PLC memiliki karakteristik:

4. Kokoh dan dirancang untuk tahan terhadap getaran, suhu, kelembaban dan kebisingan.
5. Antarmuka untuk input dan output telah tersedia secara built-in di dalamnya.
6. Mudah diprogram dan menggunakan sebuah bahasa pemrograman yang mudah dipahami, yang sebagian besar berkaitan dengan operasi-operasi logika dan penyambungan.

Perangkat PLC pertama dikembangkan pada tahun 1969. Dewasa ini PLC secara luas digunakan dan telah dikembangkan dari unit-unit kecil yang berdiri sendiri (*self-contained*) yang hanya mampu menangani sekitar 20 input/output menjadi sistem-sistem modular yang dapat menangani input/output dalam jumlah besar, menangani input/output analog maupun digital, dan melaksanakan mode-mode kontrol proporsional-integral-derivatif.



**Gambar 2.22 Sistem PLC**

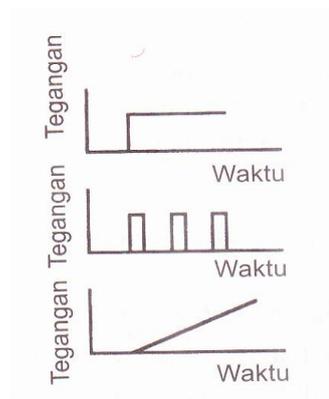
(Sumber: Bolton, William. 2004. "Programmable Logic Controller (PLC) Sebuah Pengantar". Edisi Ketiga. Penerbit Erlangga)

---

### 2.8.1 Hardware PLC

Umumnya, sebuah sistem PLC memiliki lima komponen dasar. Komponen-komponen ini adalah unit prosesor, memori, unit catu daya, bagian antarmuka input/output dan perangkat pemrograman.

1. *Unit prosesor* atau *central processing unit* (unit pengolahan pusat) (*CPU*) adalah unit yang berisi mikroprosesor yang menginterpretasikan sinyal-sinyal input dan melaksanakan tindakan-tindakan pengontrolan, sesuai dengan program yang tersimpan di dalam memori, lalu mengkomunikasikan keputusan-keputusan yang diambilnya sebagai sinyal-sinyal kontrol ke antarmuka output.
  2. *Unit catu daya* diperlukan untuk mengkonversikan tegangan AC sumber menjadi tegangan DC (5V) yang dibutuhkan oleh prosesor dan rangkaian-rangkaian di dalam modul-modul antarmuka input dan output.
  3. *Perangkat pemrograman* dipergunakan untuk memasukkan program yang dibutuhkan ke dalam memori. Program tersebut dibuat dengan menggunakan perangkat ini dan kemudian dipindahkan ke dalam unit memori PLC.
  4. *Unit memori* adalah tempat di mana program yang digunakan untuk melaksanakan tindakan-tindakan pengontrolan oleh mikroprosesor disimpan.
  5. Bagian input dan output adalah antarmuka di mana prosesor menerima informasi dari dan mengkomunikasikan informasi kontrol ke perangkat-perangkat eksternal. Sinyal-sinyal input, oleh karenanya dapat berasal dari saklar-saklar, Sinyal-sinyal output mungkin diberikan pada kumparan-kumparan starter motor, katup-katup solenoida dll. Perangkat-perangkat input dan output dapat digolongkan menjadi perangkat-perangkat yang menghasilkan sinyal-sinyal diskrit atau digital, dan yang menghasilkan sinyal-sinyal analog (**Gambar 2.23**).
-



**Gambar 2.23 Sinyal Diskrit, Digital dan Analog**

(Sumber: Bolton, William. 2004. “Programmable Logic Controller (PLC) Sebuah Pengantar”. Edisi Ketiga. Penerbit Erlangga)

Perangkat-perangkat yang menghasilkan sinyal-sinyal diskrit atau sinyal-sinyal digital adalah perangkat-perangkat yang hanya mengindikasikan kondisi ‘mati’ (*off*) atau ‘hidup’ (*on*). Sehingga saklar adalah sebuah perangkat yang menghasilkan sinyal diskrit, yaitu ada tegangan atau tidak ada tegangan. Perangkat-perangkat *digital* pada dasarnya dapat dipandang sebagai perangkat-perangkat diskrit yang menghasilkan serangkaian sinyal ‘mati’-‘hidup’. Perangkat-perangkat analog menghasilkan sinyal-sinyal yang amplitudonya sebanding dengan nilai variabel yang dipantau. Sebagai contoh, sensor suhu akan menghasilkan tegangan yang nilainya sebanding dengan suhu.

### 2.8.2 Ladder Diagram PLC Siemens

Dalam PLC Siemens, terdapat beberapa instruksi fungsi yang dapat kita gunakan untuk membantu kita dalam membuat suatu program, antara lain :

#### 1. Bit Logic

Instruksi Bit *Logic* bekerja dengan dua keadaan, yaitu “1” atau “0”. *Logic* “1” menandakan aktif dan *logic* “0” menandakan tidak aktif. Berikut ini macam-macam fungsi instruksi bit *logic* :

##### a. Normally Open Contact

Instruksi *normally open* digunakan apabila kita ingin memasukkan input yang keadaan normalnya adalah terbuka.

**Simbol**

&lt;address&gt;

--| |--

**Gambar 2.24 Ladder Diagram Normally Open****Penjelasan :**

*Address* di atas dimulai dari I0.0 sampai I65535.7 ataupun juga dapat diisi alamat instruksi lainnya, misalnya *Counter*. *Normally open contact* akan menjadi tertutup (*Closed*) ketika nilai bit dari *address* bernilai “1”.

**b. Normally Closed Contact**

Instruksi *normally close* digunakan apabila kita ingin memasukkan input yang keadaan normalnya adalah tertutup.

**Simbol**

&lt;address&gt;

--| / |--

**Gambar 2.25 Ladder Diagram Normally Close****Penjelasan :**

*Address* diatas dimulai dari I0.0 sampai I65535.7 ataupun juga dapat diisi alamat instruksi lainnya, misalnya *Counter*. Pada *normally closed contact*, saat *address* bernilai “0” saklar sudah tertutup sehingga nilai pengoperasiannya bernilai “1”, sedangkan untuk pemberian logic “1” pada saklar akan membuat saklar menjadi terbuka sehingga nilai akhir pengoperasiannya bernilai “0”.

**c. Invert Power Flow**

Instruksi *inverting* digunakan apabila kita ingin membalik suatu keadaan, misalnya dari “0” menjadi “1” atau dari “1” menjadi “0”.

**Simbol**

&lt;address&gt;

--| NOT |--

**Gambar 2.26 Ladder Diagram Inverting**

**Penjelasan :**

Fungsi dari *ladder diagram* gerbang **NOT** adalah sebagai pembalik nilai sinyal dari *input*.

**d. Output Coil**

Instruksi di atas digunakan apabila kita ingin memberikan sebuah *output*.

**Simbol**

&lt;address&gt;

--( )

**Gambar 2.27 Ladder Diagram Output Coil****Penjelasan :**

Instruksi tersebut dipasang pada akhir *network* dan statusnya dipengaruhi oleh instruksi-instruksi yang ada di depannya. *Address* diatas dimulai dari Q0.0 sampai Q65535.7.

**e. Midline Output**

Instruksi di atas digunakan apabila kita ingin membuat *output* yang terletak di tengah.

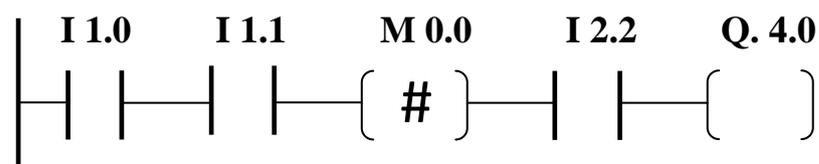
**Simbol**

&lt;address&gt;

--( # )—

**Gambar 2.28 Ladder Diagram Midline Coil****Penjelasan :**

Instruksi tersebut dipasang pada pertengahan *network* dan statusnya dipengaruhi oleh instruksi-instruksi yang ada di depannya. *Address* di atas dimulai dari M0.0 sampai M65535.7.

**Contoh aplikasi :****Gambar 2.29 Contoh aplikasi Midline Coil**

**Penjelasan :**

Pada *ladder* di atas, status *output* M0.0 akan aktif bila kedua saklar I1.0 dan I1.1 aktif. Sedangkan status *output* Q4.0 akan aktif bila ketiga saklar I1.0, I1.1 dan I1.2 aktif.

**f. Reset Coil**

Instruksi *reset* digunakan apabila kita ingin *mereset* status sebuah bit, baik keluaran (*output*) maupun *timer* ataupun *counter*.

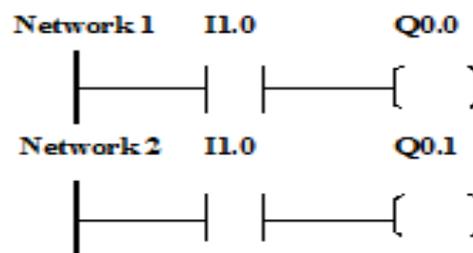
**Simbol**

&lt;address&gt;

--( R )

**Gambar 2.30 Ladder Diagram Reset Coil****Penjelasan :**

Instruksi tersebut dipasang pada akhir *network* dan statusnya dipengaruhi oleh instruksi-instruksi yang ada di depannya. *Address* diatas diisi dengan *address* dari *output/timer/counter* yang ingin kita *reset*.

**Contoh aplikasi :****Gambar 2.31 Contoh aplikasi Reset Coil****Penjelasan :**

*Output* Q0.0 akan aktif dan bernilai “1” jika nilai *input* I1.0 bernilai “1”. Jika *inputan* I1.1 aktif atau bernilai “1” maka *reset* akan aktif dan akan mereset alamat Q1.0 sehingga nilainya berubah menjadi “0” meskipun *input* I1.0 yang mengaktifkannya masih aktif.

### g. Set Coil

Instruksi *reset* digunakan bila kita ingin mengeset status sebuah bit, baik keluaran (*output*) maupun *timer* ataupun *counter*.

#### Simbol

<address>

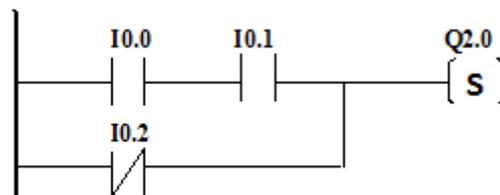
---( S )

Gambar 2.32 Ladder Diagram Set Coil

#### Penjelasan :

Instruksi tersebut dipasang pada akhir *network* dan statusnya dipengaruhi oleh instruksi-instruksi yang ada di depannya. *Address* diatas diisi dengan *address* dari *output/timer/counter* yang ingin kita *set*.

#### Contoh Aplikasi :



Gambar 2.33 Contoh aplikasi Set Coil

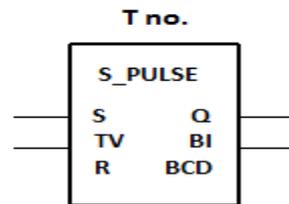
#### Penjelasan :

Pada contoh diatas, *output* Q4.0 akan di *set* bila saklar I0.0 dan I0.1 berlogic "0". *Output* Q4.0 tersebut akan aktif walaupun saklar yang ada di depannya masih terbuka.

## 2. Timers

Timer merupakan instruksi yang berfungsi memberikan waktu tunda (*delay*). Dengan adanya *timers*, kita dapat mengatur kapan suatu *output* harus aktif setelah kita berikan *input*. Selain itu kita juga dapat mengatur seberapa lama *output* tersebut harus aktif. Berdasarkan cara kerjanya, *timers* dibagi dalam beberapa macam, antara lain :

a. S\_PULSE



Gambar 2.34 *Timer S\_PULSE*

T no. = no indikasi *timer*

S = *input awal*

TV = nilai *timer*

R = *reset*

BI = nilai *timer* dengan format interger

BCD = nilai *timer* dengan format BCD

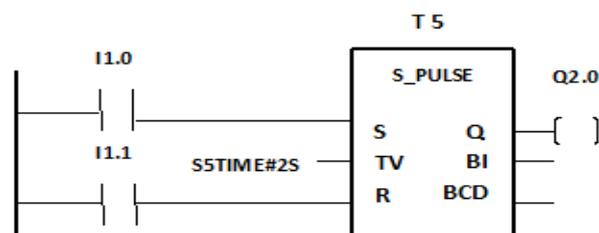
Q = status keluaran *timer*

**Penjelasan :**

Timers akan bekerja ketika input awal (S) bernilai “1”, dan lamanya menghitung (*delay time*) sesuai dengan nilai pada TV. Ketika *timers* menghitung dan nilai *input awal* (S) berubah menjadi “0” maka hitungan *timer* akan berhenti dan hitungannya akan kembali ke awal saat nilai S menjadi “1”. Nilai sinyal *output* (Q) akan bernilai “1” selama *timer* menghitung, ketika *timer* selesai menghitung nilai Q akan berubah menjadi “0”. *Timer* akan di-*reset* (*timer* bernilai “0”) ketika nilai *Reset* (R) bernilai “1”.

Nilai *timer* dapat juga menggunakan BI yaitu kode biner atau dengan BCD yaitu kode BCD.

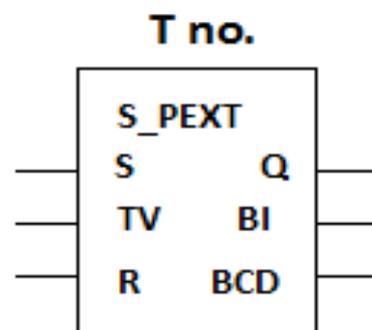
**Contoh aplikasi :**



Gambar 2.35 Aplikasi *Timer S\_PULSE*

**Penjelasan :**

Jika sinyal *input* (I1.0) bernilai “1” maka *timer* T5 akan memulai hitungannya. *Timer* akan menghitung selama 2 detik selama nilai *input* (I1.0) bernilai “1”. Jika nilai-nilai *input* (I1.0) berubah dari “1” ke “0” maka hitungan *timer* akan berhenti. Jika nilai *input* (I1.1) bernilai “1” saat *timer* menghitung maka hitungan akan berhenti dan *timer* T5 bernilai “0” (*di-reset*). Nilai *output* (Q2.0) bernilai “1” ketika *timer* menghitung, jika telah selesai menghitung nilai *output* akan menjadi “0” seperti saat *di-reset*.

**b. S\_TEXT (Extended Pulse Timer)**

**Gambar 2.36 Timer S\_PEXT**

T.no = no indikasi *timer*

S = *input* awal

TV = nilai *timer*

R = *reset*

BI = nilai *timer* dengan format interger

BCD = nilai *timer* dengan format BCD

Q = status keluaran *timer*

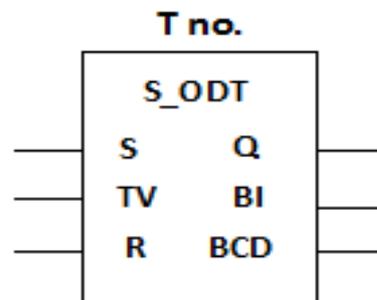
**Penjelasan :**

*Timer* akan bekerja ketika *input* awal (S) bernilai “1”. *Timer* akan terus menghitung sampai nilai *timer* (TV) habis walaupun nilai *input* awal (S) berubah nilai menjadi “0” ketika ditengah-tengah hitungan.

Ketika *timer* menghitung dan nilai *input* awal (S) berubah menjadi “0” dan berubah lagi menjadi “1” maka hitungan akan dimulai dari awal lagi. Nilai sinyal *output* (Q) akan bernilai “1” selama *timer* menghitung, ketika *timer* selesai

menghitung nilai Q akan berubah menjadi “0”. *Timer* akan di-*reset* (*timer* bernilai “0”) ketika nilai *Reset* (R) bernilai “1”. Nilai *timer* dapat juga menggunakan BI yaitu kode biner atau dengan BCD yaitu kode BCD.

c. **S\_ODT** (*On Delay Timer*)



**Gambar 2.37** *Timer S\_ODT*

T no. = no indikasi *timer*

S = *input* awal

TV = nilai *timer*

R = *reset*

BI = nilai *timer* dengan format interger

BCD = nilai *timer* denmgan format BCD

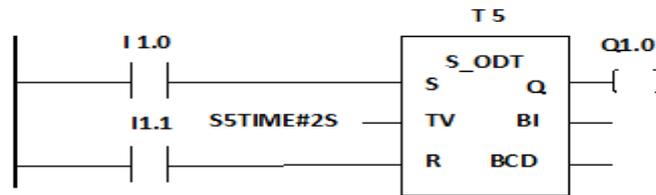
Q = status keluaran *timer*

**Penjelasan :**

*Timer* akan bekerja selama *input* awal (S) selalu berniai “1”, jika saat menghitung (S) berubah dari nilai “1” menjadi “0” maka hitungan *timer* akan berhenti. Ketika *timer* menghitung dan nilai *input* awal (S) berubah menjadi “0” dan berubah lagi menjadi “1” maka hitungan akan dimulai dari awal lagi. Nilai sinyal *output* (Q) saat *timer* menghitung adalah “0”, jika telah selesai menghitung nilai Q berubah menjadi “1”.

*Timer* akan di-*reset* (*timer* bernilai “0”) ketika nilai *Reset* (R) bernilai “1”. Nilai *timer* dapat juga menggunakan BI yaitu kode biner atau dengan BCD yaitu kode BCD.

**Contoh Aplikasi :**

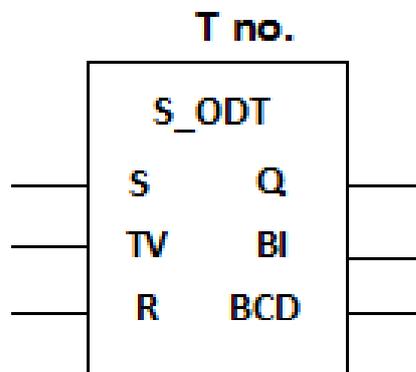


**Gambar 2.38 Aplikasi Timer S\_ODT**

**Penjelasan :**

Jika sinyal *input* (I1.0) bernilai “1” maka *timer* T5 akan memulai hitungannya. *Timer* akan menghitung selama 2 detik walaupun nilai *input* (I1.0) telah berubah menjadi “0”. Jika nilai *input* (I1.0) berubah dari “0” menjadi “1” sebelum hitungan *timer* selesai maka *timer* akan menghitung lagi dari awal. Jika nilai *input* (I1.1) bernilai “1” saat *timer* menghitung maka hitungan akan berhenti dan *timer* T5 bernilai “0” (di-*reset*). Nilai *output* (Q1.0) bernilai “0” ketika *timer* menghitung, jika telah selesai menghitung nilai *output* akan menjadi “1”.

**d. S\_OFFDT (Off Delay Time)**



**Gambar 2.39 Timer S\_OFFDT**

T no = no indikasi *timer*

S = *input* awal

TV = nilai *timer*

R = *reset*

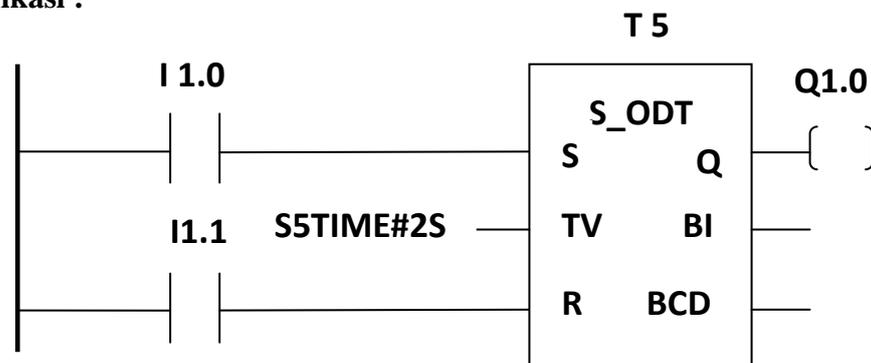
BI = nilai *timer* dengan format interger

BCD = nilai *timer* dengan format BCD

Q = status *output timer*

**Penjelasan :**

*Timer* akan bekerja ketika *input* awal (S) bernilai “1”. *Timer* akan terus menghitung sampai nilai *timer* (TV) habis walaupun nilai *input* awal (S) berubah nilai menjadi “0” ketika ditengah-tengah hitungan. Ketika *timer* menghitung dan nilai *input* awal (S) berubah menjadi “0” dan berubah lagi menjadi “1” maka hitungan dimulai dari awal lagi. Nilai sinyal *output* (Q) akan bernilai “1” selama *timer* menghitung, ketika *timer* selesai menghitung nilai Q akan berubah menjadi “0”. *Timer* akan di-*reset* (*timer* bernilai “0”) ketika nilai *Reset* (R) bernilai “1”. Nilai *timer* dapat juga menggunakan BI yaitu kode biner atau dengan BCD yaitu kode BCD.

**Contoh aplikasi :****Gambar 2.40 Aplikasi *Timer* S\_OFFTS****Penjelasan :**

Jika sinyal *input* (I1.0) bernilai “1” maka *timer* T5 akan memulai hitungannya. *Timer* akan menghitung selama 2 detik walaupun nilai *input* (I1.0) telah berubah menjadi “0”. Jika nilai *input* (I1.0) berubah dari “0” menjadi “1” sebelum hitungan *timer* selesai maka *timer* akan menghitung lagi dari awal. Jika nilai *input* (I1.1) bernilai “1” saat *timer* menghitung maka hitungan akan berhenti dan *timer* T5 bernilai “0” (di-*reset*). Nilai *output* (Q1.0) bernilai “0” ketika *timer* menghitung, jika telah selesai menghitung nilai *output* akan menjadi “0” seperti saat direset.

### 2.8.2.1 PLC SIEMENS LOGO! TYPE OBA7

Pada bagian teori kali ini, saya akan membahas PLC SIEMENS LOGO! TYPE OBA7. Tipe PLC ini digunakan pada rancangan lift 3 lantai project kami dan berfungsi sebagai pengontrolnya. Untuk penjelasannya dapat dilihat dibawah.



**Gambar 2.41 PLC Siemens Logo! Type OBA7**

PLC yang digunakan adalah merk SIEMENS LOGO! TYPE OBA7 yang dibuat oleh pabrikan SIEMENS. Software yang digunakan untuk PLC ini adalah LOGO SOFT 7 yang menggunakan bahasa *ladder diagram* atau bisa juga menggunakan *function block diagram*. PLC ini juga memiliki layar yang dapat digunakan untuk melihat maupun mengganti program yang telah diinput ke dalam PLC ini. PLC ini juga memiliki data *backup* yang dilakukan oleh SIMATIC *memory card*. Komunikasi yang digunakan adalah jaringan *Ethernet* dari komputer ke perangkat PLC ini. PLC ini memiliki *input power supply* PLC dari 200-240 VAC dan *Output power supply* sebesar 24 VDC. PLC ini memiliki jumlah *input 8 port* yang terdiri dari analog dan digital dan memiliki *output 4 relay normally open*.