

BAB II KERANGKA TEORITIS

2.1 *Reflective Micro Sensor*

Sensor yang digunakan pada sistem ini adalah *reflektive micro sensor* yang mempunyai jarak deteksi maksimal 5mm. Sensor ini bisa beroperasi pada tegangan 5V sampai dengan 24V DC. Sensor ini sudah dilengkapi dengan *transmitter* dan *recivier* yang sudah dikemas sedemikian rupa untuk memudahkan pengguna. *Output* dari sensor ini adalah digital yaitu logika "1" untuk *high* dan logika "0" untuk *low*.

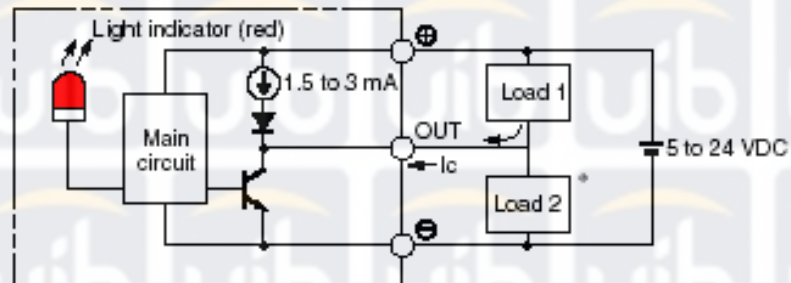
Gambar 2.1.
Reflective micro sensor



Sumber : katalog OMRON

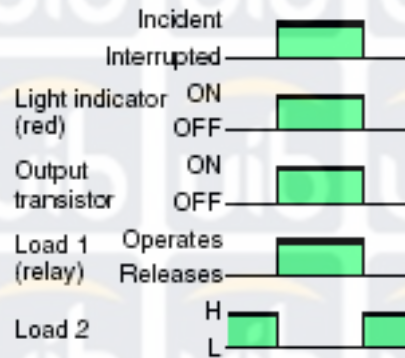
Untuk lebih jelasnya bisa dilihat pada gambar 2.2 merupakan blok diagram dari *reflektive micro sensor*. Dan pada gambar 2.3 adalah *timing chart* untuk sensor ini.

Gambar 2.2.
Blok diagram *reflective micro sensor*



Sumber : katalog OMRON

Gambar 2.3.
Timing chart reflective micro sensor



Sumber : katalog OMRON

Pada blok diagram diatas dapat dijelaskan bahwa saat tegangan diberikan pada sensor ini, maka akan mengaktifkan *main circuit* sensor. Didalam *main circuit* ini terdapat *transmitter* dan *receiver*, pada saat *transmitter* mengirimkan sinar dan terdapat benda didepan *transmitter* tersebut dalam jarak 1 sampai 5 mm maka sinar itu dipantulkan ke *receiver* dan akan mengaktifkan lampu indikator sensor dan juga memberikan tegangan ke transistor.

Pada saat *basis* transistor mendapatkan tegangan maka kaki transistor tersebut seolah-olah *short* (hubung singkat), dengan demikian *output* dari sensor ini *low*. Dan apabila tidak ada benda maka *output* dari sensor *high*. Apabila *output* yang dibutuhkan adalah negatif maka penyambungan kabelnya dapat dilakukan seperti pada *load 1*. Dan apabila yang diinginkan *output* dari sensor positif maka dapat dilakukan penyambungan seperti pada *load 2*. Akan tetapi jika disambung seperti *load 2* maka logikanya akan kebalikan dari *load 1*.

2.2 Rotary Encoder

Rotary encoder adalah alat penghitung putaran suatu motor dan mengeluarkan sinyal tiga macam yaitu *phase A*, *phase B* dan *phase Z*. Didalam *rotary encoder* ini terdapat *transmitter* dan *receiver* seperti halnya sensor yang lain. Bentuk fisik dari *rotary encoder* ini bisa dilihat pada gambar 2.6. *Rotary* ini dapat menghitung atau memberi sinyal dalam 1 putaran sebanyak 200 sinyal. Banyaknya sinyal yang dikeluarkan dinamakan resolusi. Untuk sinyal yang dikeluarkan oleh *rotary encoder* ini bisa dihitung dengan menggunakan rumus:

$$\text{Maximum electrical response speed (rpm)} = \frac{\text{Maximum response frequency}}{\text{Resolution}} \times 60$$

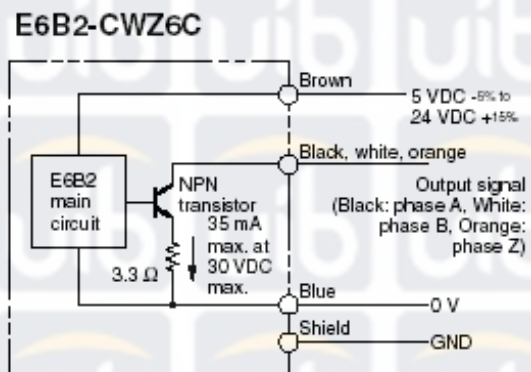
Gambar 2.4.
Rotary Encoder



Sumber : katalog OMRON

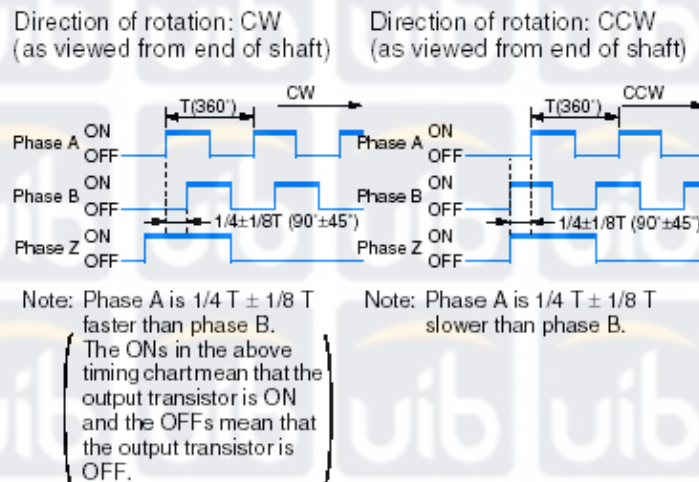
Pada gambar 2.5 merupakan blok diagram dari *rotary encoder*. Dan pada gambar 2.6 adalah *timing chart* untuk tiga *output rotary encoder* jika diputar searah jarum jam (CW) dan kebalikan arah jarum jam (CCW). *Output* dari *rotary* ini adalah *digital* yaitu logika “1” untuk *high* dan logika “0” untuk *low*.

Gambar 2.5.
Blok diagram *rotary encoder*



Sumber : katalog OMRON

Gambar 2.6.
Timing chart output rotary encoder



Sumber : katalog OMRON

2.3 *Reed Switch*

Reed switch ini biasanya dipakai apabila kita menggunakan silinder.

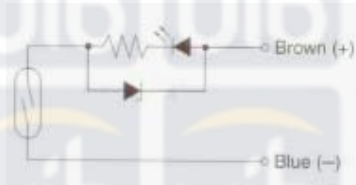
Tetapi silinder yang dipakai adalah tipe yang menggunakan magnet. Magnet tersebut terdapat pada piston silinder. Cara kerja dari *reed switch* ini adalah saklar yang ada di dalamnya akan aktif apabila dekat dengan magnet (magnet yang ada di dalam silinder).

Gambar 2.7.
Reed switch



Sumber : katalog Koganei

Gambar 2.8.
Blok diagram reed switch



Sumber : katalog Koganei

2.4 *Push button*

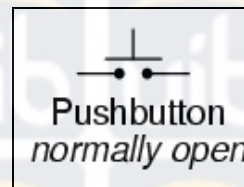
Kebanyakan *switch* yang ada adalah *push button*, karena juga penggunaannya yang sangat luas dalam otomotif dan aplikasi peralatan elektronik.

Ada dua tipe *push button*, yaitu

- a. *Momentary Pushbutton switch* aktif ketika tombol ditekan dan tidak aktif ketika tombol dilepas.
- b. *Maintained Pushbutton switch* aktif ketika ditekan dan tetap aktif ketika dilepas, untuk mematikan saklar harus ditekan untuk yang kedua kali. Karena alasan inilah biasanya *switch* ini dikatakan sebagai *push-push switch* atau *on/off switch*

Kontak pada *switch* dapat dibagi jadi dua tipe, *Normaly Open (N/O)* dan *Normaly Close (N/C)*, ketika *Switch* dalam kondisi tidak aktif maka kontak N/O akan terbuka (*Non – Conducting*) dan N/C akan tertutup (*Conducting*)

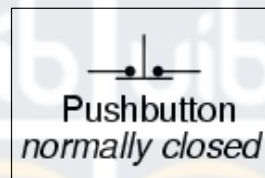
Gambar 2.9.
Simbol push button normally open



Sumber : katalog IDEC

Gambar 2.9 untuk simbol skematik dari *normally open push button* dan untuk simbol skematik dari *normally close push button* ini dapat dilihat pada gambar 2.10.

Gambar 2.10.
Simbol push button normally close

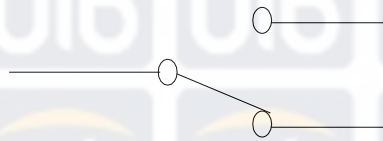


Sumber : katalog IDEC

2.5 Toggle switch

Cara kerja *toggle switch* hampir sama dengan *maintained Pushbutton switch*, cuma bentuknya saja yang berbeda. Perbedaan cara kerjanya adalah *maintained pushbutton switch* ditekan sedangkan *toggle switch* digeser.

Gambar 2.11.
Simbol toggle switch

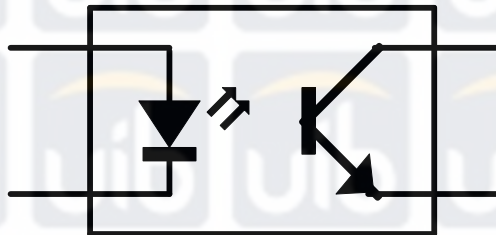


Sumber : katalog IDEC

2.6 Optocoupler

Optocoupler adalah salah satu contoh dari alat-alat optoelektronika yaitu teknologi yang menggabungkan optika dan elektronika, dimana alatnya adalah bidang yang menyajikan beberapa alat yang didasarkan atas perilaku persambungan P-N. *Optocoupler* terbentuk dari gabungan LED dan photo transistor. Lihat gambar 2.9.

Gambar 2.12
Optocoupler dengan LED dan Phototransistor



Sumber: Prinsip-prinsip elektronika (Edisi Ketiga), Erlangga.

LED (*Light Emitting Diode*) atau dioda pemancar cahaya, terbentuk dari material semikonduktor tipe P dan tipe N yang dibatasi oleh *junction* yang terbentuk dalam *single kristal*. Pada *junction* terjadi kombinasi dari *free electron* N dan *free hole* P, sehingga pada sisi N akan bermuatan positif sedangkan pada posisi P akan bermuatan negatif. Pada dioda tegangan maju, elektron bebas melintasi persambungan dan jatuh kedalam lubang (*hole*). Pada saat elektron ini jatuh dari tingkat energi yang lebih tinggi ke tingkat energi yang lebih rendah, ia memancarkan energi. Pada dioda-dioda biasa, energi ini keluar dalam bentuk panas, tetapi pada LED energi memancar sebagai cahaya.

Sebuah transistor dengan basis yang terbuka mempunyai arus kolektor yang kecil yang terdiri dari panas yang dihasilkan oleh pembawa muatan minoritas dan arus bocor permukaan. Dengan membuka sedikit sambungan (*junction*) kolektor untuk diberi cahaya, pabrik dapat membuat phototransistor, suatu transistor yang lebih sensitif terhadap cahaya. Gambar 2.12 diatas menunjukkan sebuah LED menggerakkan sebuah phototransistor. *Optocoupler* diatas mempunyai sensitif yang tinggi. Jika sebuah tegangan diberikan ke LED, maka setiap perubahan pada tegangan menghasilkan arus pada LED, yang merubah arus melalui phototransistor. Sebaliknya, ini menghasilkan suatu perubahan tegangan pada terminal kolektor emitor. Karena itu, tegangan sinyal dikopel dari rangkaian *input* ke rangkaian *output*. Keuntungan besar dari *optocoupler* adalah adanya isolasi listrik antara rangkaian *input* dan rangkaian *ouput*.

2.7 *Microcontroller*

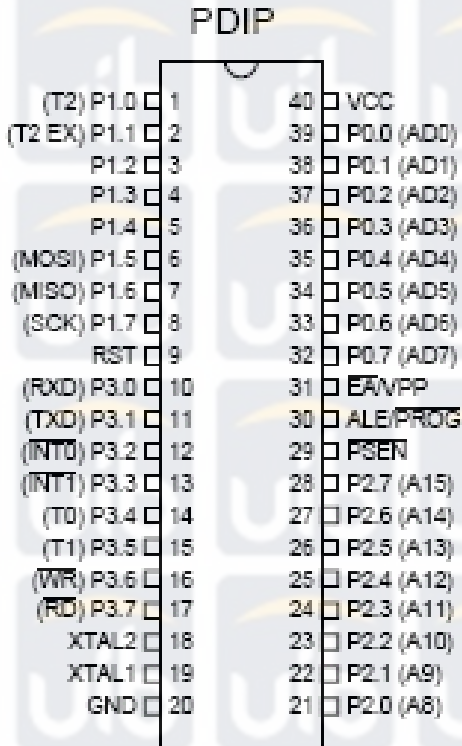
Microcontroller merupakan sebuah komponen elektronika yang didalamnya terdapat rangkaian *microprocessor*, memori (*RAM / ROM*) dan *I/O*. Rangkaian tersebut dikemas dalam satu chip yang biasa disebut *Single Chip Microcomputer*.

Pada *Microcontroller* sudah terdapat komponen - komponen *microprocessor* dengan *bus – bus internal* yang saling berhubungan. Komponen – komponen tersebut adalah *RAM*, *ROM*, *Timer*, Komponen *I/O paralel* dan *serial*, dan *interrupt controller*. Sistem *interrupt* merupakan keunggulan dari *Microcontroller*. Sistem ini dapat menaikan respon semangat *eksternal* (*Interrupt*) diwaktu yang nyata. Perangkat tersebut dapat melakukan hubungan *switching* yang cepat dan menunda satu proses ketika ada respon eksekusi yang lain.

2.8 **AT89S52**

AT89S52 adalah salah satu keluarga dari *Single Chip Micro Computer* (*microcontroller*), dan merupakan salah satu jenis *microcontroller* CMOS 8 bit dengan *performa* yang tinggi dan disipasi daya yang rendah. Ini merupakan *flash* memori 8 Kilobyte dengan daya tahan 1000 kali penulisan dan penghapusan (*write / erase*) yang memiliki 40 pin. Komponen ini *compatible* dengan AT80C51 *standart*, mulai dari instruksi di set sampai pin out. ATMEL AT89S52 merupakan suatu *microcontroller* yang kuat dengan fleksibilitas yang tinggi dan suatu solusi hemat biaya bagi penggunanya.

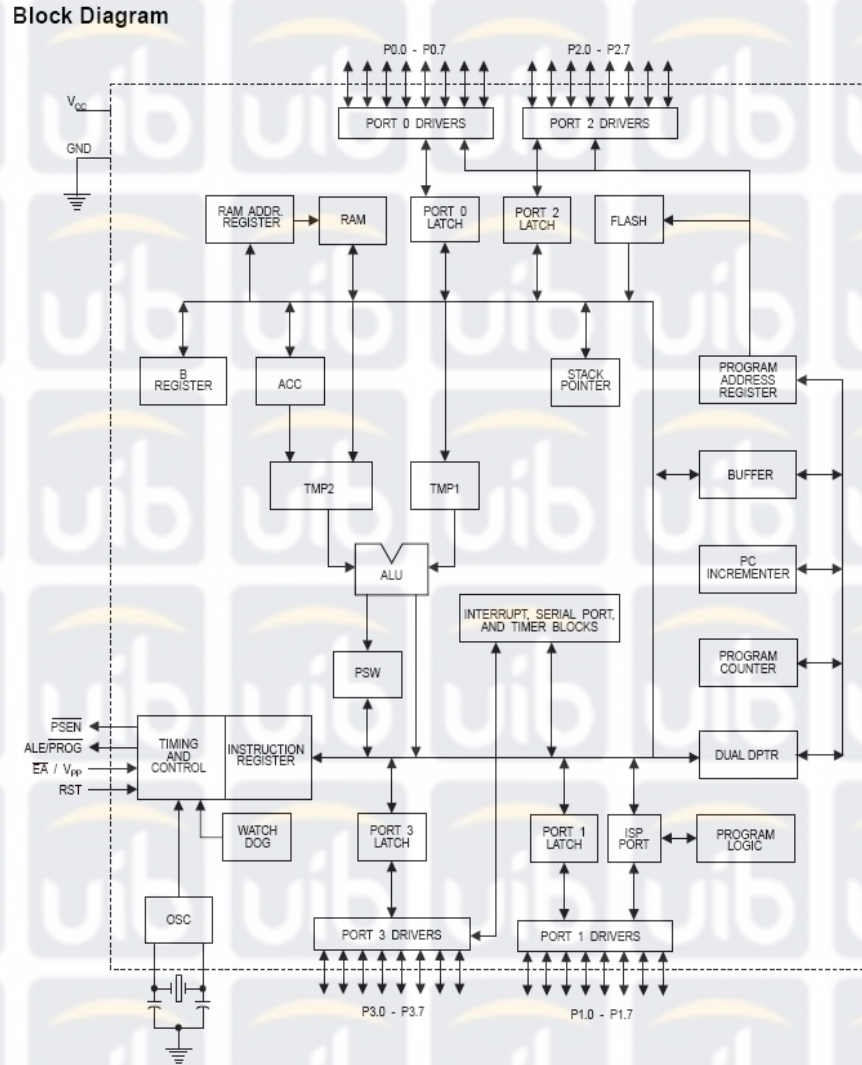
Gambar 2.13
Konfigurasi pin AT89S52



Sumber : Data Sekunder ATMEL 1997

Gambar 2.13. diatas menunjukkan bentuk dari IC AT89S52, dan menunjukkan pin – pin IC dengan fungsinya masing – masing. Dari konfigurasi di atas kita akan melihat blok diagram dari IC AT89S52, seperti yang terlihat pada gambar 2.14 berikut.

Gambar 2.14
Block Diagram AT89S52



Sumber : data sekunder ATMEL 1997

Blok diagram tersebut menjelaskan cara kerja dari AT89S52, dengan fungsi konfigurasi. Penjelasan dari pin – pin AT89S52 adalah seperti berikut.

2.8.1 Pin 1 – 8: Port 1

Pin 1-8 merupakan saluran I/O dua arah dan sebagai saluran alamat pada saat pemrograman dan verifikasi.

2.8.2 Pin 9: *Reset*

Pin 9 merupakan masukan *reset* (aktif tinggi), sehingga pulsa transisi dari rendah ke tinggi mengembalikan posisi semula (*reset*) pada mikrokontroler.

2.8.3 Pin 10 – 17: *Port 3*

Pin 10-17 merupakan saluran / bus I/O 8 bit dua arah dengan *internal pull-ups* yang memiliki fungsi pengganti. Bila fungsi pengganti tidak dipakai, maka dapat digunakan sebagai *port* paralel 8 bit serbaguna. Juga berfungsi sebagai sinyal kontrol pada saat proses pemrograman dan verifikasi. Adapun fungsi penggantian seperti pada tabel 2.1 berikut ini:

Tabel 2.1
Fungsi pengganti dari *port 3*

Bit	Nama	Fungsi Alternatif
P3.0	RXD	Untuk menerima data <i>port serial</i>
P3.1	TXD	Untuk mengirim data <i>port serial</i>
P3.2	INT0	Interupsi eksternal 0
P3.3	INT1	Interupsi eksternal 1
P3.4	T0	<i>Input</i> Eksternal waktu/pencacah 0
P3.5	T1	<i>Input</i> Eksternal waktu/pencacah 1
P3.6	WR	Jalur menulis memori data eksternal
P3.7	RD	Jalur membaca memori data eksternal

Sumber: Modul Batara Elektrindo (2001).

2.8.4 Pin 18 – 19: *XTAL1* dan *XTAL2*

Pin 18-19 merupakan masukan ke osilator berpenguat tinggi. Mikrokontroler ini memiliki seluruh rangkaian osilator yang diperlukan pada bagian yang sama (*on chip*) kecuali rangkaian kristal yang mengendalikan frekuensi osilator. Karenanya pin 18 dan pin 19 sangat perlu dihubungkan dengan kristal. Selain itu *XTAL 1* dapat juga sebagai

input untuk *inverting oscillator amplifier* dan *input* ke rangkaian *internal clock* sedangkan XTAL 2 merupakan *output* dari *inverting oscillator amplifier*

2.8.5 Pin 20: GND

Pin 20 berfungsi untuk *grounding*.

2.8.6 Pin 21 – 28: Port 2

Pin 21-28 merupakan saluran / *bus* I/O 8 bit dua arah dengan *internal pull-ups*. Saat pengambilan data dari program memori eksternal atau selama mengakses data memori eksternal yang menggunakan alamat 16 bit (MOVX @ DPTR), *port 2* berfungsi sebagai saluran / *bus* alamat tinggi (A8 – A15). Sedangkan pada saat mengakses ke data memori eksternal yang menggunakan alamat 8 bit (MOVX @ R1), *port 2* mengeluarkan isi dari P2 pada *Special Function Register*

2.8.7 Pin 29: PSEN

Program Store Enable (PSEN) merupakan sinyal pengontrol untuk mengakses program memori eksternal masuk ke dalam bus selama proses pemberian / pengambilan instruksi (*fetching*).

2.8.8 Pin 30: ALE

Address Latch Enable (ALE) / PROG merupakan penahan alamat memori eksternal (pada *port 1*) selama mengakses ke memori eksternal. pin ini juga sebagai pulsa / sinyal *input* pemograman (PROG) selama proses pemograman.

2.8.9 Pin 31: EA

Pin 31 merupakan *external Access Enable* (EA) merupakan sinyal kontrol untuk pembacaan memori program. Apabila di set rendah (L) maka mikrokontroller akan melaksanakan seluruh instruksi dari memori program eksternal, sedangkan apabila diset tinggi (H) maka mikrokontroller akan melaksanakan instruksi dari memori program internal ketika isi program *counter* kurang dari 4096. ini juga berfungsi sebagai tegangan pemrograman ($V_{PP} = +12V$) selama proses pemrograman

2.8.10 Pin 32 – 39: Port 0

Pin 32-39 merupakan saluran / bus I/O 8 bit *open collector Multiplexer* antara bus alamat rendah dan bus data, selama adanya akses ke memori program eksternal. Pada saat proses pemrograman dan verifikasi *port 0* digunakan sebagai saluran / bus data. *External pull-ups* diperlukan selama proses verifikasi.

2.8.11 Pin 40: Vcc

Pin 40 merupakan pin untuk sumber positif tegangan.

2.9 Bahasa Assembler

Bahasa Pemrograman yang digunakan untuk mengendalikan mikrokontroller adalah bahasa *assembler*, yang menggunakan empat teknik pengalamatan, yaitu:

2.9.1 Pengalamatan Langsung

Teknik pengalamatan langsung dilakukan dengan memberikan nilai ke suatu *register* secara langsung. Untuk melaksanakan teknik pengalamatan langsung digunakan tanda “#”.

2.9.2 Pengalamatan Tak Langsung

Teknik pengalamatan tidak langsung menunjuk ke sebuah *register* yang berisi lokasi alamat memori yang akan digunakan dalam operasi. Lokasi yang sebenarnya tergantung pada isi *register* saat instruksi dijalankan. Untuk melaksanakan pengalamatan tidak langsung digunakan simbol “@”.

2.9.3 Pengalamatan Bit

Teknik pengalamatan bit adalah penunjukkan alamat lokasi bit, baik dalam RAM *internal* atau perangkat keras. Untuk melakukan pengalamatan bit digunakan simbol titik (.), misalnya *FLAGS.3*, *40.5*, *21H.5*, dan *ACC.7*.

Sebagai contoh instruksi:

- a. *SETB TR1* (mengaktifkan *TR1* / timer 1 on) dan
- b. *SETB 88H.6* (mengaktifkan bit 6 pada lokasi *88H* / timer 1 on).

2.9.4 Pengalamatan Kode

Pengalamatan kode merupakan pengalamatan ketika *operand* alamat dari instruksi *jump* dan *call* (*ACALL*, *JMP*, *LJMP*, dan *LCALL*). Biasanya *operand* tersebut akan menunjuk kesuatu alamat yang telah diberi label sebelumnya.

2.10 *Visual Basic*

Visual Basic (atau sering disingkat VB) adalah perangkat lunak untuk menyusun program aplikasi yang bekerja dalam lingkungan sistem operasi *windows*.

Dengan *visual basic* maka kemampuan *windows* bisa dimanfaatkan secara optimal. Dengan kelebihan yang dimiliki, bisa dirasakan kemudahan menyusun program aplikasi dengan tampilan grafis yang menawan dalam waktu yang relatif singkat. Berikut ini adalah bahasa pemrograman *visual basic* :

2.10.1 *Variable*

Variable adalah tempat penyimpanan data, yang merupakan *pointer* yang menunjuk alamat memori fisik tertentu. Dalam melakukan pemrograman selalu dibutuhkan tempat penyimpanan data, misalnya untuk menampung data hasil perhitungan, menampung data hasil pembacaan *register* atau lainnya.

a. Tipe data dan deklarasi *variable*

Dalam penggunaannya *variable* harus mempunyai nama dan tipe data tertentu. nama *variable* menunjuk pada suatu tempat dalam memori komputer, sedangkan tipe data mengontrol besarnya memori yang disediakan untuk *variable* tersebut.

Tabel 2.2
Tipe data pada *visual basic*

Tipe Data	Ukuran Byte	Range
Integer	2 byte	-32.768 s/d 32.767
Long Integer	4 byte	-2.147.483.648 s/d 2.147.483.647
Single-precision floating point	4 byte	-3,402823 E38 sampai 3,402823 E38
Double-Precision	8 byte	Negatif (-1,79769313486232E308 s/d -4,94065645841247E-324) Positif (4,94065645841247E-324 s/d 1,79769313486232E308)
Currency	8 byte	-933337203685477,5808 sampai 922337203685477,5807
String	1 byte per character	0 s/d 65535 karakter
Boolean	2 byte	True atau False
Date	8 byte	1 Januari 100 s/d 31 Desember 9999
Byte	1 byte	0 s/d 255
Variant	16 byte untuk angka; 22 byte + 1 byte per karakter (untuk string)	Semua tipe data

Sumber : Data primer (2008)

b. Array

Array dapat digunakan pada sekumpulan *variable* dengan nama dan tipe data yang sama. Untuk mengakses *variable* tertentu dalam *array* harus menggunakan *indeks*. Data yang disimpan dalam *array* tersebut selanjutnya disebut dengan elemen. Cara pendeklarasian *array* adalah seperti pendeklarasian *variable*, hanya saja harus mengikutkan jumlah elemennya.

2.10.2 Kontrol Program

Kekuatan Pemrograman adalah terletak pada kontrol. Dengan kontrol program maka akan mengendalikan alur eksekusi program dan menentukan keputusan apa yang harus dikerjakan oleh program. Kontrol pada VB meliputi kontrol pertimbangan kondisi dan keputusan, kontrol perulangan, dan kontrol penyaluran alternatif.

2.10.3 Prosedur

Program bisa dibuat lebih mudah dengan memecah program menjadi blok-blok komponen yang lebih kecil. Ini dinamakan prosedur. Prosedur sangat berguna ketika program yang dibuat sering menggunakan berulang-ulang tugas yang sama atau bermaksud membagikannya pada program lain. Dengan prosedur akan lebih mudah program dibagi beberapa step karena kita dapat melakukan pengujian prosedur daripada pengujian seluruh program.

2.10.4 Sub Prosedur

Sub prosedur adalah blok kode yang dijalankan sebagai tanggapan atas terbentuknya *even*, baik *even* itu merupakan *even* pemanggilan dari prosedur lain maupun *even* yang terjadi dari pemakai program.

Setiap kali prosedur dipanggil, blok pernyataan yang ada diantara *Sub* dan *End Sub* akan dijalankan. Argumen untuk sebuah *prosedure* mirip dengan deklarasi *variable*, yaitu pemakaian suatu nilai yang melewati pemanggilan *prosedur*.

Untuk memanggilnya bisa ditulis nama prosedur disertai argumennya dengan tipe data yang harus sama. Setiap kali prosedur diatas dipanggil, prosedur ini akan menentukan angka apa yang dilewatkan pada argumen dan akan mencetaknya pada *form*.

2.10.5 Fungsi manipulasi *string*

Fungsi – fungsi dalam kelompok ini biasa digunakan untuk tugas – tugas yang berhubungan dengan pemrosesan *string*

Tabel 2.3
Tabel Fungsi manipulasi *string*

Nama Fungsi	Keterangan
Len	Menentukan panjang sebuah string
Ucase	Mengubah huruf string menjadi huruf besar
Lcase	Mengubah huruf string menjadi huruf kecil
Right	Menghasilkan karakter yang dihitung dari sisi kanan string
Left	Menghasilkan karakter yang dihitung dari sisi kiri string
Mid	Menghasilkan karakter yang dihitung dari tengah – tengah string dimulai dari titik awal tertentu
String	Menghasilkan karakter string
Asc	Menghasilkan kode ASCII dari huruf tertentu
Chr	Menghasilkan karakter untuk kode ASCII tertentu
Trim	Menghasilkan karakter dengan menghilangkan karakter spacenya
Str	Menghasilkan karakter yang merepresentasikan angka numerik
InStr	Mencari titik awal dari sebuah string dalam string besar
Val	Menghasilkan angka numerik yang terkandung dalam string

Sumber : Data primer (2008)

2.10.6 Operator

Operator digunakan pada *visual basic* untuk memanipulasi data maupun untuk melakukan perhitungan. Operator pada *visual basic* dapat

dikelompokkan menjadi tiga kelompok yaitu: operator matematik, operator perbandingan, dan operator logika.

a. Operator Matematik

Operator matematik digunakan untuk melakukan operasi perhitungan matematik. Dalam VB jika suatu ekspresi mempunyai operator lebih dari satu kategori, maka operator matematik yang akan dievaluasi terlebih dulu dilanjutkan dengan operator perbandingan, dan yang terakhir operator logika. Tabel berikut adalah operator matematik dimulai dari hirarki yang paling tinggi ke yang paling rendah.

Tabel 2.4
Tabel operator matematik

Operator	Operasi
^	Pemangkatan
-	Tanda Negatif
*, /	Perkalian & Pembagian
\	Pembagian Integer
Mod	Modulus (Sisa Pembagian)
+, -	Penambahan & Pengurangan
&	Penggabungan String

Sumber : Data primer (2008)

b. Operator Perbandingan

Operator perbandingan digunakan untuk membandingkan suatu ekspresi dengan ekspresi lain yang menghasilkan nilai boolean (*false* atau *true*). Tabel berikut adalah operator perbandingan :

Tabel 2.5
Tabel operator perbandingan

Operator	Operasi
=	Sama dengan
<>	Tidak sama dengan
<	Lebih Kecil
>	Lebih Besar
<=	Lebih kecil atau sama dengan
>=	Lebih besar atau sama dengan

Sumber : Data primer (2008)

c. Operator Logika

Operator logika digunakan untuk mengekspresikan satu atau lebih ekspresi logika yang akan menghasilkan nilai boolean. Tabel berikut adalah operator logika dimulai dari hirarki yang paling tinggi ke paling rendah beserta tabel kebenarannya.

Tabel 2.6
Tabel operator logika

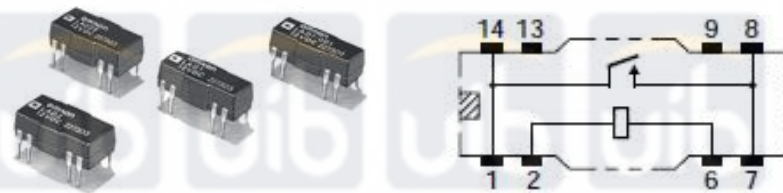
Operator	Operasi	Tabel Kebenaran Operasi	
		Operan	Hasil
Not	Akan menghasilkan nilai kebalikan dari nilai ekspresi Boolean	Not True	False
		Not False	True
OR	Akan menghasilkan nilai True jika salah satu ekspresi operannya bernilai logika True	True OR True	True
		True OR False	True
		False OR True	True
		False OR False	False
And	Akan menghasilkan nilai True hanya jika kedua ekspresi operannya bernilai logika True	True And True	True
		True And False	False
		False And True	False
		False And False	False

Sumber: Data Primer (2008).

2.11 Relay

Relay adalah sebuah saklar elektromekanis yang didalamnya terdiri dari sebuah *coil* dengan inti besi yang berfungsi sebagai elektromagnet yang akan menarik saklar mekanis sehingga saklar tersebut akan terhubung (*closed*). Seperti tampak gambar 2.15 dibawah ini, prinsip kerjanya sama seperti sebuah *switch*. *Relay* kontaknya akan tertutup bila kumparan (*coil*)nya dialiri arus listrik (untuk *NO relay*). *Relay* berfungsi untuk pengaturan ON/OFF atau menghidupkan /mematikan alat.

Gambar 2.15
Relay yang digunakan dan Skematik.



Sumber: Data Primer (2008).

2.12 Indicator Lamps

Indicator lamps berfungsi menginformasikan kepada operator bahwa power telah di nyalakan atau mengindikasikan status dari mesin yang sedang beroperasi. Gambar dari indikator lamp ditunjukkan pada gambar 2.16

Gambar 2.16.
Indicators lamp



Sumber : Katalog Idec

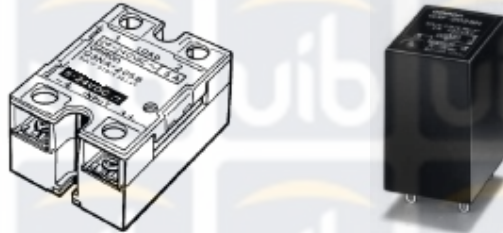
Warna dari *indicator lamps* biasanya merah (untuk *safety critical indicator* = *power is on, machine is running, fault has occurred*), hijau (untuk kondisi *safe* = *Power motor is off, brake are on, etc*), kuning (untuk kondisi penting tetapi tidak berbahaya = *fluid getting low, machine paused, machine warming up, manual, etc*). Kadang apabila dalam keadaan yang benar benar *emergency* biasanya digunakan lampu *flash* (*on – off* berulang ulang) untuk menarik operator.

2.13 Solid state relay (SSR)

Solid state relay (SSR) berfungsi hampir sama dengan *relay* biasa tapi SSR ini tidak menimbulkan *spark* atau bunga api pada keluarannya, sedangkan *relay* terjadi *spark* pada keluarannya. SSR ini ada berbagai macam yaitu *input DC output DC*, *input DC output AC*, *input AC input DC*, dan *input AC output AC*.

Tapi yang dipakai disini yaitu *input DC output DC* dan *input DC output AC*.

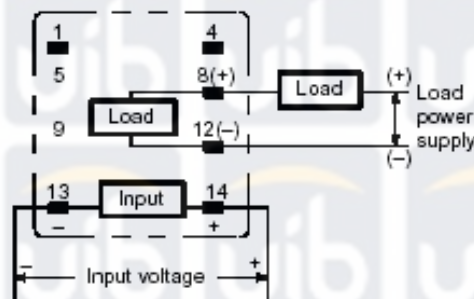
Gambar 2.17.
Bentuk SSR



Sumber : katalog OMRON

Penyambungan SSR ini sangatlah mudah, untuk SSR *input DC output DC* kaki no 13 dan 14 tidak boleh terbalik polaritasnya. Yaitu kaki no 13 masuk *negative (-)* dan kaki 14 tegangan *positif (+)*, untuk output kaki 12 diberi tegangan *negative (-)* sedangkan kaki 8 diberikan ke *negative* beban/ *load*. Untuk lebih jelas bisa dilihat pada gambar 2.18 dibawah ini.

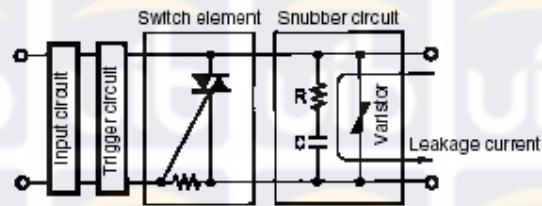
Gambar 2.18.
Penyambungan SSR DC



Sumber : katalog OMRON

Akan tetapi untuk penyambungan SSR *input AC output AC* tidaklah terlalu rumit karena jika terbalik tidak akan merusak alat itu sendiri, asalkan tidak terbalik antara *input* dan *output*.

Gambar 2.19.
Diagram blok SSR AC



Sumber : katalog OMRON