BAB 2

LANDASAN TEORI

2.1 Fuzzy Logic

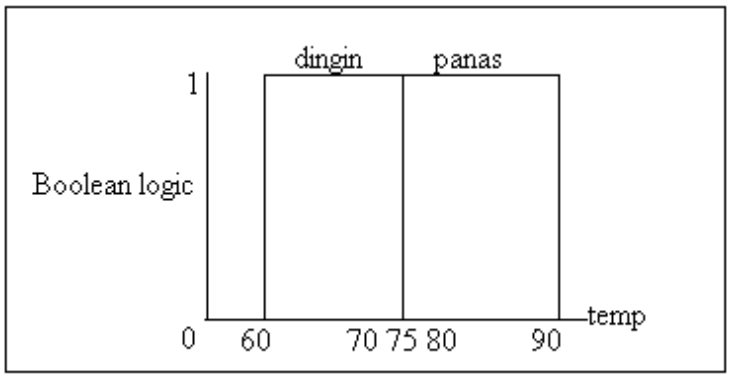
Teori *fuzzy* pertama kali diperkenalkan oleh Prof. Lofti Zadeh dari Universitas California USA, pada tahun 1965. Logika *fuzzy* berbeda dengan logika digital biasa (*Boolean logic*), dimana pada logika digital biasa hanya dapat mengenal dua keadaan yaitu : Ya / Tidak atau *ON* / *OFF* atau *High* / *Low*. Sedangkan logika *fuzzy* meniru cara berpikir manusia dengan menggunakan konsep sifat kesamaran suatu nilai. Dengan *fuzzy logic* maka nilai yang diberikan dapat berawal dari nol lalu berubah secara kontiniu sampai nilai satu. Teknologi *fuzzy* ini sudah sangat berkembang pesat dan banyak digunakan pada pemakaian pengaturan lalulintas, sistem transmisi otomatis, industri, peralatan rumah tangga, dan lain-lain.

2.1.1 Perbedaan Fuzzy Logic dan Boolean Logic

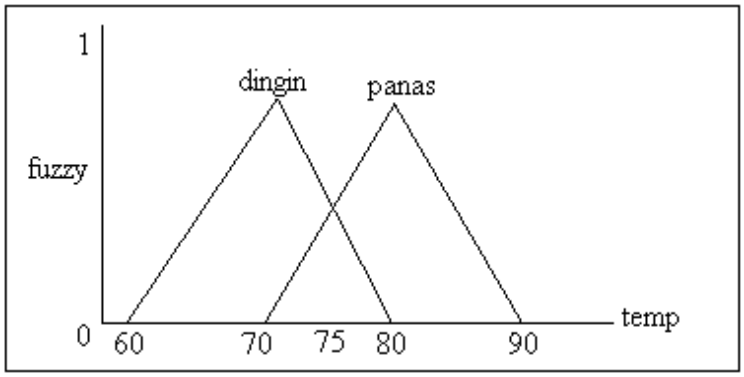
Dengan menggunakan *fuzzy logic* maka hasil yang dapat diperoleh akan lebih presisi dibandingkan dengan *boolean logic*, hal ini disebabkan karena *Boolean logic* hanya dapat mengenal 0 dan 1, sementara dengan *fuzzy* kita dapat mengenal antara 0 sampai 1. perbedaan antara keduanya akan di ilustrasikan pada gambar 2.1 dan gambar 2.2 dibawah.

Dari gambar 2.1 dibawah, pada saat suhu berada pada 75° maka sistem yang pertama akan menghadapi kendala dalam menentukan statusnya karena batas kondisi dingin lebih kecil dari 75 dan kondisi panas lebih besar dari 75. Dengan menggunakan

fuzzy logic maka suhu 75° dapat dinyatakan dengan 0,5 panas dan 0,5 dingin. Pengambilan nilai 0,5 berasal dari proses fuzzifikasi.



Gambar 2.1 Boolean Logic



Gambar 2.2 Fuzzy Logic

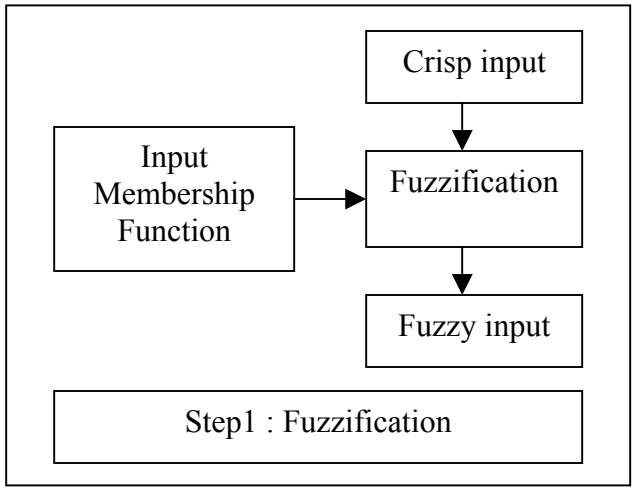
.

2.1.2 Fuzzy Control

Fuzzy controller sebenarnya memiliki prinsip kerja yang sama dengan sistem konvensional, dimana ia akan menerima suatu *input*, kemudian melakukan kalkulasi dan selanjutnya membangkitkan *output* berdasarkan hasil perhitungan yang diperoleh. Jadi prosedur yang diperlukan dalam pengendalian *fuzzy control* dapat dibagi menjadi tiga secara garis besar seperti yang telah diperkenalkan oleh Ebrahim Mamdani pada akhir tahun 1970-an. Tiga tahap tersebut adalah sebagai berikut :

1. *Fuzzification.*
2. *Rule Evaluation.*
3. *Defuzzification.*

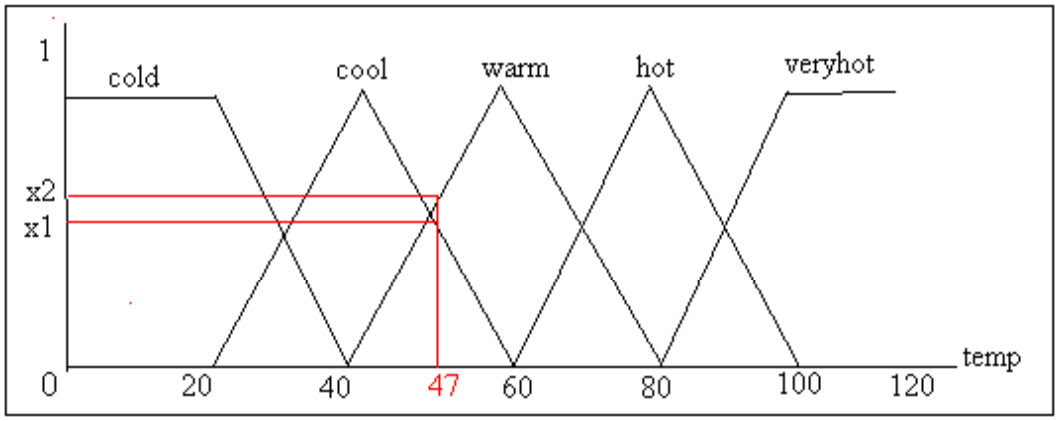
Tahap 1 : Secara garis besar tahap pertama ini berfungsi untuk mengubah suatu besaran analog menjadi *fuzzy input*. Untuk blok diagram dari tahap fuzzifikasi ini dapat dilihat pada gambar 2.3.



Gambar 2.3 Blok Diagram Fuzzification

Penjelasan dari proses diatas adalah sebagai berikut : Suatu besaran analog dimasukkan sebagai *input* (*crisp input*), lalu *input* tersebut dimasukkan pada batas *scope*/domain sehingga *input* tersebut dapat dinyatakan dengan label (dingin, panas, cepat, lambat, dll) dari *membership function*. *Membership function* ini biasanya dinamakan *membership function input*. Dari *membership function* inilah kita dapat mengetahui berapa *degree of membership function*-nya (derajat keanggotaan). Contoh dari proses fuzzifikasi akan diilustrasikan dalam gambar 2.4 dibawah ini. Sebuah sistem *fuzzy* untuk mengukur suhu

mempunyai 5 buah *membership function* yang mempunyai label : *cold, cool, warm, hot, veryhot*. Kemudian *input* yang diperoleh dari *crisp input* adalah 47, maka proses pengambilan *fuzzy input* dari *crisp input* sebagai berikut :

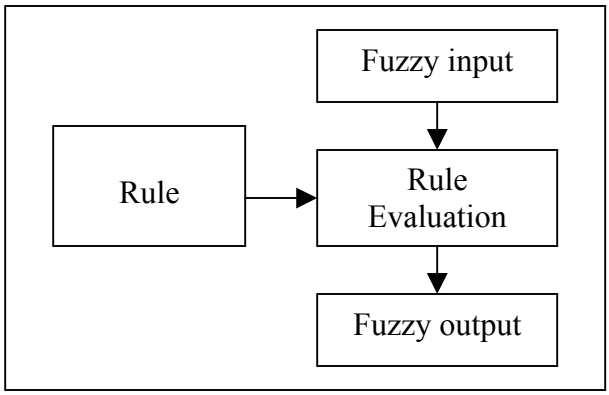


Gambar 2.4 Proses Perubahan dari Crisp Input Menjadi Fuzzy Input

Dari proses diatas diperoleh 2 *fuzzy input* yang masing-masing adalah *cool* (X2) dan *warm* (X1). Yang menentukan kesensitifan dari suatu sistem adalah banyaknya *membership function*, semakin banyak maka semakin sensitif. Sensitif disini maksudnya adalah memiliki respon yang cepat, sehingga jika terjadi sedikit perubahan saja pada *input* maka sistem akan merespon dan memberikan suatu *output* yang lain.

Tahap 2 : Proses ini berfungsi untuk melakukan perhitungan dalam mencari suatu nilai *fuzzy output* dari *fuzzy input*. Prosesnya adalah sebagai berikut : suatu nilai *fuzzy input* yang diperoleh dari proses fuzzifikasi kemudian dimasukkan kedalam sebuah *rule* yang telah ditentukan sebelumnya untuk dijadikan sebuah *fuzzy output*. Pada saat penentuan *fuzzy rule* haruslah sangat diperhatikan dengan benar, karena pada bagian inilah yang menentukan seberapa pandaikah sistem tersebut. Jika terjadi kesalahan dalam

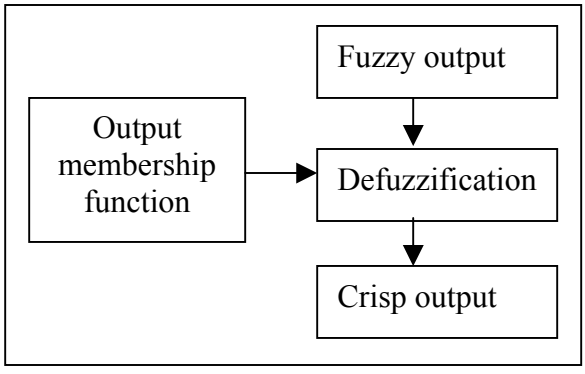
menentukan *fuzzy rule* dapat mengakibatkan sistem yang akan dikontrol menjadi kacau. Gambar blok diagram dari tahap ini dapat dilihat pada gambar 2.5 :



Gambar 2.5 Blok Diagram Rule Evaluation

Pada sistem ini *rule* yang digunakan adalah menggunakan suatu hubungan sebab akibat (*If / then*) dengan contoh penulisan seperti : *if condition is* X *then action is* Y.

Tahap 3 : merupakan tahap terakhir yang dilakukan. Tujuan dari proses ini adalah untuk mengubah keluaran *fuzzy* yang diperoleh dari hasil perhitungan tahap 2 menjadi keluaran nyata berupa pengontrolan aktuator. Untuk blok diagram dari tahap ini dapat dilihat pada gambar 2.6 dibawah.

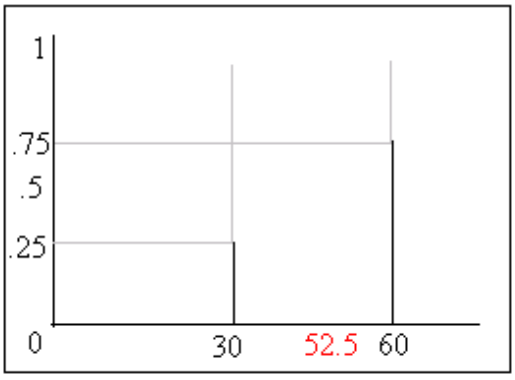


Gambar 2.6 Proses Defuzzification

Prosesnya adalah sebagai berikut : suatu nilai *fuzzy output* yang diperoleh dari *rule evaluation* dimasukkan kedalam suatu *membership function output*. Bentuk bangun yang digunakan pada *membership function output* disini sebagai contoh digunakan bentuk *singelton* (garis lurus vertikal keatas), seperti yang ditunjukkan pada gambar 2.7. besar nilai *fuzzy output* dinyatakan sebagai *degree of membership function output*. Nilai-nilai tersebut dimasukkan kedalam suatu rumus yang dinamakan COG (*Center Of Gravity*) sehingga diperoleh hasil akhir yang disebut *crisp output* (keluaran eksaknya dalam bentuk analog).

Rumus COG yang digunakan pada proses ini adalah sebagai berikut :

$$\text{Crisp output (Y)} = \frac{\sum i(\text{fuzzy output}_i) \text{ X (Singelton posisiton on x axis}_i)}{\sum i(\text{fuzzy output}_i)} \qquad (2.1)$$



Gambar 2.7 Bentuk Fuzzy Output

Dari gambar diatas maka perhitungan *crisp output*-nya :

$$\frac{(0 \text{ x } 0) + (0.25 \text{ x } 30) + (0.75 \text{ x } 60)}{0 + 0.25 + 0.75} = 52.5 \qquad (2.2)$$

2.2 Mikrokontroler MCS-51

Mikrokontroler 89C51 merupakan produk dari ATMEL dimana 89C51 ini masih termasuk keluarga dari 8051 karena memiliki konfigurasi pin dan instruksi set yang sama.

Karakteristik dari 89C51 adalah sebagai berikut :

- 128 x 8 *bit* RAM internal,
- Mempunyai 4 *Kbyte Flash Programmable* and *Erasable Read Only Memory,*
- Mempunyai 64 *Kbyte* untuk alamat program dan data eksternal,
- Maksimum data dan I/O port sebesar 64 *Kbyte*,
- Mempunyai 2 buah 16 *bit timer counter*,
- Mempunyai 4 buah I/O port 8 bit,
- Mempunyai fasilitas komunikasi serial *Full duplex universal asynchronous receiver Transmitter* (UART),
- Mempunyai 3 level penguncian memory program.

Pada penelitian ini digunakan modul DT-51 sebagai modul untuk minimum sistemnya. Dimana spesifikasi dari DT-51 adalah sebagai berikut :

- Berbasiskan mikrokontroler 89C51.
- Serial port interface standar RS-232 untuk komunikasi antara komputer dengan DT-51.
- 8 *Kbytes non-volatile memory* (EEPROM) untuk menyimpan program dan data.
- 128 *byte* Internal RAM

- 4 *Kbytes Flash Programmable and Erasable Read Only Memory*.
- 4 port *input output* (I/0) dengan kapasitas 8 *bit* tiap portnya.
- 2 buah 16 *bit Timer / Counter*
- Port LCD untuk keperluan *display* (tampilan)
- Konektor ekspansi.
- Selain komponen tersebut masih terdapat fungsi-fungsi lain pada DT51, antara lain : RS-232 serial port, *Programmable Peripheral Interface* (PPI), serta LCD port.

  - **Programmable Peripheral Interface (PPI)**

    PPI berfungsi sebagai I/O *Expander* yang dapat diprogram. PPI yang digunakan mempunyai 24 bit jalur *input output* yang dapat dihubungkan dengan peralatan atau *device* lain. 24 bit I/O ini dibagi menjadi 3 port yaitu Port A, Port B, dan Port C.

  - **TTL ⇔ RS 232**

    DT51 berkomunikasi dengan PC secara serial. Proses *download* dan *debugging* dilakukan melalui serial port. 89C51 mempunyai sebuah serial port dengan level standar TTL. Supaya bisa berkomunikasi dengan serial port PC (COM1/COM2) yang mempunyai level standar RS-232, maka diperlukan konverter level TTL ⇔RS 232.

  - **LCD Port**

    LCD (*Liquid Crystal Display*) Port ini disiapkan untuk men-drive LCD melalui Port 1 mikrokontroler 89C51.

  - **Memori Eksternal**

Selain PEROM dan internal RAM yang terdapat pada 89C51, DT51 juga mempunyai memori eksternal berjenis EEPROM dengan kapasitas *8 Kbytes* untuk menyimpan *user* program atau data yang di-*download* dari PC.

2.2.1 Address Space MCS-51

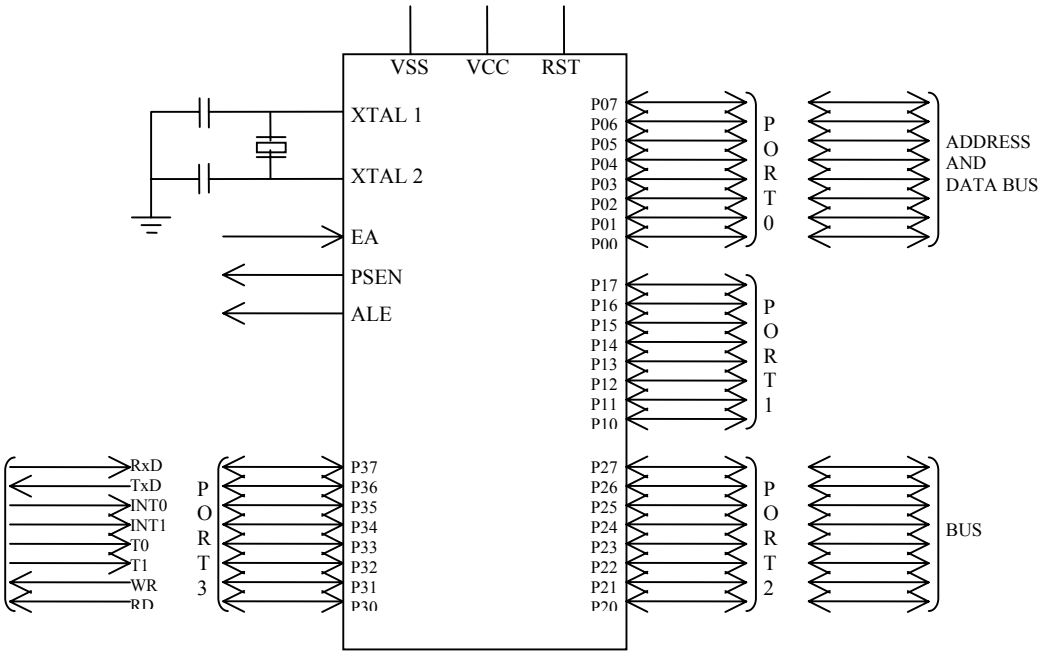
Pada mikorokontroler MCS51 secara garis besar terdiri atas dua jenis memory yaitu RAM untuk menyimpan data dan ROM untuk menyimpan alamat. RAM yang digunakan pada mikorokontroler itu memiliki kapasitas sebesar 128 *byte*, RAM ini disebut dengan RAM internal karena memory ini berada didalam mikrokontroler itu sendiri. Terkadang karena diperlukan kapasitas RAM yang lebih besar untuk penyimpanan data, maka dapat digunakan RAM eksternal. Dimana pengaksesan antara mikrokontroler dan RAM eksternal dilakukan melalui pin control RD dan WR. ROM internal yang digunakan pada 89C51 ini sebesar 4 *Kbyte*. Sama seperti pada RAM jika kapasitas yang disediakan tidak mencukupi dapat digunakan ROM eksternal. ROM eksternal yang biasa digunakan pada MCS51 ini adalah EPROM. Untuk menentukan ROM mana yang diakses ditentukan dari masukan pada pin EA, jika untuk pengaksesan ROM internal adalah dengan menghubungkan pin EA ke Vcc sehingga dapat mengakses alamat dari 0000h-1FFFh. Sedangkan untuk mengakses ROM eksternal adalah dengan menghubungkan pin EA ke *ground* sehingga dapat mengakses alamat mulai dari 2000h sampai FFFFh. Dengan jalur alamat (*addressing*) sampai 16 bit dan jalur data sebesar 8 *bit*, maka sistem minimum ini dapat melakukan pengalamatan sampai dengan 64 *Kbyte* (2^{16}X2^{8}) baik untuk RAM atau ROM.

2.2.2 Konfigurasi Pin 89C51

Untuk gambar konfigurasi pin pada 89C51 dapat dilihat pada gambar 2.8. Fungsi-fungsi konfigurasi dari pin 89C51 adalah sebagai berikut :

Vcc : Merupakan tegangan catu daya yang diperlukan oleh mikrokontroler untuk beroperasi.

RST (*Reset*) : Pin ini digunakan untuk mereset *processor* bila diberi logika 1. *reset* diperlukan untuk proses insialisasi internal, seperti mengisi register-register dengan nilai tertentu dan melakukan instruksi *JUMP* ke alamat awal dari programnya, yaitu 0000h.



Gambar 2.8 Konfigurasi Pin 89C51

Port : Pada MCS 51 terdapat empat buah port yaitu port 0 sampai dengan port 3, dimana setiap portnya tersusun lagi atas 8 port pin. Port 0 terdiri atas P0.0 sampai

dengan P0.7 demikian juga dengan port 1, 2, dan 3. Pada MCS 51 port 0 dan port 2 berfungsi sebagai address dan data bus. Untuk keterangan port adalah sebagai berikut :

1. Port 0 (P0.0….P0.7) sebagai address dan data bus orde rendah, dan juga berfungsi sebagai I/O *bi-directional* 8 bit, dimana setiap bitnya dapat mengendalikan 8 masukan gerbang TTL.
2. Port 1 (P1.0…...P1.7) sebagai I/O *bi-directional* 8 bit, dengan tiap bitnya dapat mengendalikan 4 masukan TTL. Port ini mempunyai rangkaian *pull up* internal.
3. Port 2 (P2.0….P2.7) sebagai I/O *bi-directional* 8 bit, dan juga sebagai *address* dan data bus orde tinggi. Setiap bitnya dapat mengendalikan 4 masukan TTL.
4. Port 3 (P3.0….P3.7) selain sebagai I/O *bi-directional* 8 bit port ini memiliki fungsi–fungsi khusus. Port ini dapat mengendalikan 4 masukan TTL. Fungsi khusus dari port 3 dapat dilihat pada tabel 2.1 dibawah ini :

Tabel 2.1 Fungsi Khusus dari Port 3 pada MCS 51

Port Pin	**Alternate Function**
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	WR (external data memory write strobe)
P3.7	RD (external data memory read strobe)

ALE (*Address Latch Enable*) : Digunakan untuk mengeluarkan pulsa dengan tujuan untuk menahan (*latch*) *low address* selama pengaksessan ke memory eksternal.

PSEN (*Program Store Enable*) : Digunakan untuk mengeluarkan pulsa pada saat proses *fetching* (pengambilan) instruksi ataupun pencarian data dalam bentuk tabel (*look-up table*) dari ROM atau EPROM.

EA (*External Access Enable*) : Sebagai penentu apakah mikrokontroler akan melaksanakan instruksi dari memory internal atau dari memory eksternal. Apabila EA diberi logika 1 maka yang diakses adalah memory eksternal, dan jika diberi logika 0 maka yang diakses adalah memory internal.

XTAL1 : Merupakan pin masukan ke rangkaian osilator internal IC.

XTAL2 : Merupakan pin keluaran ke rangkaian osilator internal IC.

2.2.3 Modus Pengalamatan

Modus pengalamatan pada mikrokontroler 89C51 sebagai berikut :

A. Pengalamatan Langsung (*Direct Addressing*)

Menggunakan alamat 8 *bit*, hanya RAM data internal dan SFR dapat diakses dengan pengalamatan langsung. Data dari operand sumber akan langsung masuk ke alamat memori yang terdapat pada operand tujuan.

Misal : MOV P1,A

Artinya : memindahkan isi akumulator ke port 1

B. Pengalamatan Tidak Langsung (*Indirect Addressing*)

Menggunakan sistem pengalamatan alamat 8 atau 16 *bit* dan dapat mengakses RAM internal maupun eksternal. Register yang dapat dipakai untuk menyimpan

alamat yaitu register R0, R1, DPL, DPH untuk pengalamatan 8 *bit* dan DPTR untuk pengalamatan 16 bit. Untuk menandakan *indirect addressing* digunakan simbol "@" di depan nama register penyimpan alamat.

Misal : MOV R0, #50H MOV DPTR, #0FF00H

MOV @R0, #0BCH MOVX@DPTR,#0ABH

Artinya : lokasi 50H pada RAM data internal diisi dengan data 0BCH, lokasi FF00H pada RAM data eksternal diisi dengan data 0ABH.

RAM eksternal hanya dapat diakses dengan pengalamatan tak langsung. Untuk transfer data dari/ke RAM eksternal, akumulator selalu menjadi sumber/tujuan dari data.

Misal : MOVX A, @Ri

Artinya : membaca RAM eksternal pada lokasi @Ri

Misal : MOVX @Ri, A

Artinya : menulis RAM eksternal pada lokasi @Ri

C. Pengalamatan Segera (*Immediate Addressing*)

Berfungsi untuk memasukkan data konstanta ke internal memori dan register. Data bilangan konstanta harus diawali dengan tanda pagar "#".

Misal : MOV A, #105

Artinya : mengisi akumulator dengan data konstan 105

D. Pengalamatan Index (*Indexed Addressing*)

Pengalamatan ini hanya dapat dipakai untuk mengalamati memori program eksternal, yaitu data yang terdapat pada ROM atau EPROM. Pengalamatan

berindeks ini berguna dalam pengaksesan tabel (*Look Up Table*) pada memori program eksternal.

Pengalamatan 16 bit dapat digunakan register basis (*Base Register*), DPTR (*Data Pointer*) atau PC (*Program Counter*).

Alamat sebenarnya merupakan jumlah dari data pada akumulator dan penunjuk basis (*Base Pointer*).

Misal : MOVC A, @A+DPTR

Artinya : mengambil data dari program memori eksternal yang alamatnya ditunjuk oleh register DPTR dan akumulator, kemudian dipindahkan ke akumulator.

2.2.4 Interupsi

Pada MCS51 secara garis besar ada tiga buah interupsi, dimana ketiga interupsi tersebut memiliki alamat interupsi vektor (*interrupt vector*) yang statis. Ketiga interupsi tersebut adalah :

1. Interupsi untuk penanganan *timer* (T0,T1).
2. Interupsi untuk komunikasi serial (UART : Rx dan Tx).
3. Interupsi untuk penanganan sinyal dari luar / *external interrupt* (INT0,INT1).

Ketiga interupsi diatas dapat diubah urutan prioritasnya sesuai dengan aplikasi, dan interupsi diatas dapat diaktifkan ataupun di-nonaktifkan. Untuk menggunakan interupsi yang harus dilakukan adalah menset bit EA pada register IE menjadi *high* (1), kemudian langkah selanjutnya adalah set bit dari instruksi yang dikehendaki pada register IE

menjadi 1, dan langkah terakhir adalah memberikan alamat *interrupt service routine* sesuai alamat *interrupt vector* tersebut.

Interrupt vector pada MCS 51 dapat dilihat pada tabel 2.2 dibawah ini :

Tabel 2.2 Interrupt Vector

No	Sumber Interupsi	Alamat Vektor	Keterangan
1	IE0	0003h	Interupsi eksternal ke-0
2	TF0	000Bh	Interupsi dari timer ke-0
3	IE1	0013h	Interupsi eksternal ke-1
4	TF1	001Bh	Interupsi dari timer ke-1
5	R1,T1	0023h	Interupsi komunikasi serial
6	TF2,EXF2	002Bh	Interupsi dari timer ke-2

Register-register yang umum berkenaan dengan menggunakan interupsi:

a. IE : *Interrupt enable register*, dapat diamati per bit.

IE.7	IE.6	IE.5	E.4	IE.3	IE.2	IE.1	IE.0
EA	-	ET2	ES	ET1	EX1	ET0	EX0

EA : *Enable All*, jika bernilai 1 maka interupsi diatur pada bit selanjutnya, jika bernilai 0 maka tidak ada interupsi yang dilayani.
ES : *Enable serial port interrupt* (1 = *enable* , 0 = *disable*).
Etn : *Enable timer* ke-n (1 = *enable* , 0 = *disable*).
Exn : *Enable External Interrupt* ke-n (1 = *enable* , 0 = *disable*).

- EA : jika EA = 0, maka semua interupsi tidak aktif. Jika EA = 1, maka interupsi akan aktif bila bit dari interupsi yang bersangkutan aktif (logika 1).
- ES : jika ES = 1, maka interupsi untuk serial port diaktifkan saat telah menerima 1 byte data atau selesai mengirim 1 byte data. Jika ES = 0, maka interupsi untuk serial port tidak diaktifkan.

- ET1 : jika ET1 = 1, maka interupsi akan dibangkitkan pada saat *timer* 1 menghitung penuh (FFFFh) atau *overflow*. Jika ET1 = 0, maka interupsi tidak dibangkitkan.
- EX1 : jika EX1 = 1, maka interupsi akan dibangkitkan pada saat pin INT1 diberi logika 0, jika EX1 = 0, maka interupsi tidak dibangkitkan.
- ET0 : jika ET0 = 0, maka interupsi akan dibangkitkan pada saat *timer* 0 menghitung penuh (FFFFh) atau *overflow*. Jika ET0 = 0, maka interupsi tidak dibangkitkan.
- EX0 : jika EX0 = 1, maka interupsi akan dibangkitkan pada saat pin INT0 diberi logika 0, jika EX0 = 0, maka interupsi tidak dibangkitkan.

b. IP : *Interrupt Priority Register*, dapat diamati per bit.

IP.7	IP.6	IP.5	IP.4	IP.3	IP.2	IP.1	IP.0
-	-	PT2	PS	PT1	PX1	PT0	PX0

PS : Level prioritas untuk serial port.
PTn : Level prioritas untuk *timer* ke-n.
PXn : Level prioritas untuk eksternal interupsi ke-n.

Interupsi yang bersangkutan akan memiliki prioritas yang lebih tinggi dari interupsi yang lainnya, apabila bit pada IP diberi logika 1. Apabila semua bit bernilai 0 (kondisi awal), maka prioritas dari masing-masing interupsi akan ditentukan oleh algoritma internal yang terdapat didalam IC. Urutan interupsi pada kondisi standar adalah sebagai berikut (dimulai dari prioritas tertinggi) :

1. IE0 : *interrupt external* 0

2. TF0 : *interrupt timer* 0
3. IE1 : *interrupt external* 1
4. TF1 : *interrupt timer* 1
5. R1 dan T1 : *interrupt serial* (UART).

2.2.5 Pewaktu (Timer)

Pada mikrokontroler 89C51, pewaktu (*timer*) dapat berfungsi sebagai pencacah (*counter*) atau sebagai pewaktu. Pada modus operasi pencacah, pencacah akan naik setiap ada sinyal masukan yang diberikan pada pin T0 (*timer 0*) dan T1 (*timer 1*). Untuk modus operasi sebagai pewaktu, pewaktu akan menghitung setiap siklus instruksi mesin (*machine cycle*). Modus operasi sebagai pewaktu atau pencacah dapat dipilih dengan memberi logika 1 atau 0 pada bit tertentu pada register TMOD (*Timer Mode Control*). Nilai untuk *timer* atau *counter* dimasukkan pada register THx (byte tinggi) dan TLx (byte rendah). Untuk memakai atau memberhentikan *timer* atau *counter* di kontrol oleh bit TRx yang terdapat pada register TCON.

Beberapa register yang penting untuk pewaktu atau pencacah, yaitu :

1. TMOD (*Timer/Counter Mode Control Register*).
2. TCON (*Timer/Counter Control Register*).

TMOD, Fungsi TMOD adalah :

- Meng-*enable* atau *disable timer/counter* saat terjadi interupsi eksternal.
- Memilih modus dari *timer/counter*.
- Memilih sumber clock untuk *timer/counter*.

Format dari TMOD adalah sebagai berikut :

MSB LSB

GATE	T/C	M1	M0	GATE	T/C	M1	M2
Timer 0							Timer 1

Keterangan :

GATE : Jika gate = 1, maka *timer / counter* akan aktif bila pin INTx berlogika 1 (aktivitas *timer / counter* dikendalikan secara *hardware*).

Jika gate = 0, maka *timer / counter* akan aktif, jika TRx berlogika 1 (aktifitas *timer / counter* dikendalikan secara *software*).

T/C : berfungsi untuk memilih modus *timer atau counter*.

Jika T/C = 1, maka akan berfungsi sebagai *timer*.

Jika T/C = 0, maka akan berfungsi sebagai *counter* .

Tabel 2.3 dibawah ini menunjukkan bahwa *timer / counter* pada mikrokontroler 89C51 memiliki 4 macam modus operasi yang dipilih dengan mengeset bit M0 dan M1 pada TMOD register, yakni :

Table 2.3 Pemilihan Modus Timer

M1	**M2**	**Modus Operasi**
0	0	Timer 13 bit
0	1	Timer/Counter 16 bit
1	0	8 bit AutoReload Timer/Counter
1	1	(Timer 0) men-stop Timer/Counter 8 bit dikontrol oleh bit Timer 0. TH0 adalah 8 bit Timer dikontrol oleh bit kontrol Timer 1
1	1	(Timer 0) men-stop Timer/Counter

TCON, Fungsi TCON adalah :

- Mengaktifkan atau tidak mengaktifkan *timer / counter*.
- Melihat status (*flag*) dari *timer / counter*.
- Mendeteksi adanya *trigger* dari interupsi eksternal.

Format TCON adalah sebagai berikut :

MSB LSB

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Keterangan :

TF1 : bit yang menandakan bahwa *timer / counter* 1 telah mencapai nilai hitungan maksimum.

TR1 : bit yang mengontrol aktifitas dari *timer / counter* 1 untuk mulai menghitung.

TF0 : bit yang menandakan bahwa *timer / counter* 0 telah mencapai nilai hitungan maksimum.

TR0 : bit yang mengontrol aktifitas dari *timer / counter* 0 untuk mulai menghitung.

IE1 : bit yang menandakan terjadinya interupsi eksternal 1.

IT1 : bit yang mengontrol cara pengaktifan interupsi eksternal 1.

IE0 : bit yang menandakan terjadinya interupsi eksternal 0.

IT0 : bit yang mengontrol cara pengaktifan interupsi eksternal 0.

2.3 Perangkat Input

Masukan (*input*) adalah rangsangan yang diterapkan ke sebuah sistem pengendalian dari sumber energi luar, dengan tujuan agar menghasilkan tanggapan tertentu dari sistem pengendalian tersebut.

2.3.1 Sensor

Sensor merupakan suatu piranti yang mengalirkan efek fisikal seperti panas, cahaya atau getaran mekanik menjadi sinyal tegangan. Beberapa jenis sensor yang sering digunakan yaitu sensor suhu, sensor kelembaban, sensor tekanan, sensor optik dan lain sebagainya. Sensor yang digunakan di sini adalah sebuah sensor suhu LM35. Beberapa ketentuan yang mempengaruhi jenis sensor :

1. Linearitas

   Sensor yang memperhatikan sinyal keluaran yang berubah secara kontinyu sebagai tanggapan terhadap masukan yang berubah secara kontinyu. Linearitas terbagi atas :

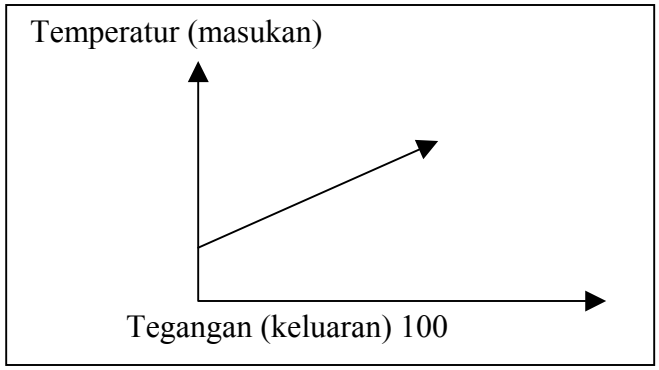
   a. Tanggapan Linear

      Masukan yang memperhatikan sinyal keluaran yang berubah secara kontinyu sehingga memudahkan kalibrasi sensor. Keluaran dari tanggapan linear dapat dilihat pada Gambar 2.9.

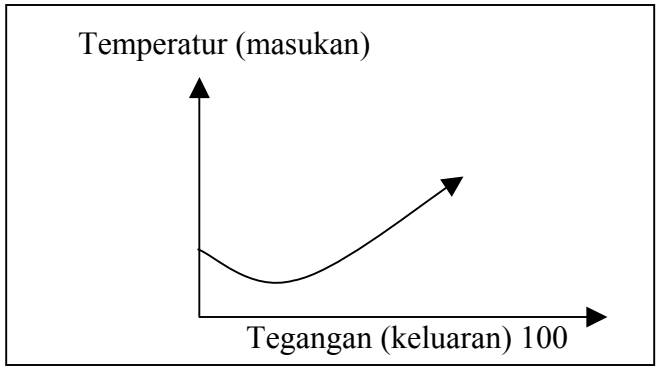
   b. Tanggapan non-linear

      Dapat mencakup jangkauan temperatur yang lebih besar, di mana tegangan keluaran akan bervariasi dalam jangkauan yang cukup kecil.

Untuk perhitungan kalibrasi agak sulit tetapi biasanya menggunakan persamaan matematika dengan kurva yang menghubungkan tegangan keluaran dengan masukan. Gambar 2.10 dibawah ini memperlihatkan keluaran dari tanggapan non-linear.



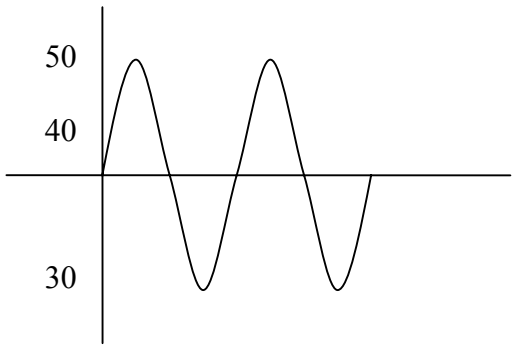
Gambar 2.9 Keluaran dari Tanggapan Linear



Gambar 2.10 Keluaran dari Tanggapan Non-Linear

2. Sensitivitas

Sensitivitas akan menunjukkan seberapa jauh kepekaan sensor terhadap kuantitas yang diukur yang dapat dinyatakan "Perubahan keluaran dibandingkan dengan unit perubahan dalam masukan".



Gambar 2.11 Temperatur yang Berubah Secara Kontinyu

Sebagai contoh, sensor suhu yang dapat dinyatakan dengan "satu mili Volt perderajat". Hal tersebut, tergantung jenis linearitas dari sensor. Apabila tanggapannya linear, maka sensitivitasnya juga akan sama dalam menjangkau pengukuran keseluruhannya. Hal tersebut dapat dilihat pada Gambar 2.11.

3. Tanggapan Frekuensi

   Tanggapan frekuensi pada sensor menunjukkan seberapa cepat tanggapannya terhadap perubahan pada masukan. Sebagai ukuran di dalam sensor ini yaitu frekuensi di dalam jumlah siklus dalam satu detik dan diberikan dalam satuan *hertz* (Hz) atau *decibel* (dB).

Beberapa kriteria penting yang dapat dijadikan acuan dalam pemilihan sensor yaitu :

a. Rentangan

   Rentangan merupakan nilai yang berkisar dari batas atas dan batas bawah dari *output* sensor. Akurasi sensor dinyatakan dalam persen terhadap rentangan, sehingga pemilihan dengan rentangan yang sangat besar akan mengurangi akurasi yang sebenarnya.

b. Akurasi

Akurasi, merumuskan perbedaan batas nilai yang diperbolehkan antara pembacaannya yang sebenarnya dari sensor terhadap nilai harga variabel yang diproses.

c. Repeatability

Repeatability, menunjukkan kemampuan ketetapan nilai yang dapat diberikan dalam beberapa kali pengukuran. *Repeatability* lebih penting dibanding akurasi dalam banyak aplikasi. Untuk mencapai tujuan *repeatability* maka kelak diperlukan sedikit penyesuaian dalam menetapkan set.

d. Lingkungan

Lingkungan, didefinisi sebagai kondisi yang diperbolehkan agar sensor dapat beroperasi.

e. Read out

Read out, menunjukkan respon sinyal dari alat ukur terhadap kondisi proses. Sebuah sensor harus dapat menghasilkan sinyal yang sesuai dengan pencatat, pengontrol, komputer ataupun alat yang menerima sinyalnya.

f. Linearitas

Linearitas, pada sebuah sinyal respon merupakan fungsi garis lurus, tidak eksponensial (logaritma). Sebuah sensor yang terbentuk dari bagian- bagian pada sistem alat ukur akan membentuk linearitas dari sistem secara keseluruhan.

2.3.1.1 Sensor Suhu

LM35 adalah sensor suhu yang berbentuk IC, yang tegangan *output-t*nya secara linear proporsional terhadap suhu celcius. Berikut adalah karakteristik dari LM35 :

- Dapat dikalibrasi dalam celcius
- Skala faktornya +10mV/°C
- Jangkauan suhu antara -55 sampai +150 °C
- Akurasinya 0,5 °C (pada +25°C)
- Beroperasi dari 4-30 V
- Arus *drain*nya kurang dari 60 µA
- Ketidaklinearan hanya sebesar ¼°C
- Impedansi *output*-nya rendah, 0, 1Ω untuk 1mA

2.3.2 Keypad

Keypad merupakan satu set saklar tombol tekan yang terhubung dalam pola X/Y agar status dari setiap tombolnya dapat dengan mudah terbaca. Sistem ini menggunakan keypad berukuran 3 baris x 4 kolom yang terdiri dari tombol numerik 0 – 9 serta tombol # dan tombol *. Untuk mengenali tombol yang ditekan maka keypad ini dibantu dengan IC key encoder tipe MM74C922. IC yang dirancang untuk keypad ukuran matrix 4 x 4 tersebut akan men-*decode input* yang berasal dari penekanan tombol *keypad* kedalam bentuk data biner.

Bila ada penekanan tombol pada *keypad*, maka kode dan baris tersebut dikodekan ke 4 bit data biner paralel dan ditampung pada *output buffer register*. Jika telah terdapat data pada *output buffer register*, maka 74C922 akan mengeluarkan pulsa

strobe melalui pin DAV (*Data Available*) sehingga kondisinya berubah dari *low* ke *high*. Data keluarannya selalu di-*latch* (tahan) sehingga memberikan waktu yang cukup bagi mikrokontroler untuk mengambil data dari keypad tersebut. Pin DAV dihubungkan ke pin INT0 pada AT89C51. Perubahan kondisi yang terjadi pada pin DAV ini mengakibatkan pin INT0 aktif dan program akan meloncat ke prosedur interupsi 0.

2.4 Perangkat Output

keluaran (*output*) adalah tanggapan sebenarnya yang diperoleh dari sebuah sistem pengendalian.

A. Aktuator

Aktuator merupakan peralatan yang dikendalikan oleh sistem, dimana cara kerjanya adalah dengan mengubah sinyal kontrol yang telah terkonversi ke masukan yang dibutuhkan elemen kontrol.

B. Heater

Heater merupakan aktuator yang berfungsi untuk mengubah energi listrik menjadi energi panas. Heater tersusun atas sebuah elemen pemanas yang diletakkan dalam sebuah tabung, dan jika tegangan diberikan pada elemen tersebut maka secara perlahan elemen tersebut akan menghasilkan energi panas sampai batas kemampuan elemen tersebut.

C. Relay

Relay adalah suatu piranti yang menggunakan magnet listrik untuk mengoperasikan seperangkat kotak. Susunan paling sederhana terdiri atas kumparan kawat penghantar yang memutari teras magnet. Bila kumparan itu dienergikan oleh arus (pada umumnya DC tetapi yang jenis AC juga ada), medan magnet yang dibangun menarik armatur berporos, memaksanya bergerak cepat kearah atas. Gerakan armatur ini dipakai melalui pengungkit, untuk menutup atau membuka kontak-kontak. Beberapa susunan kontak yang dapat dipakai, semuanya itu secara listrik terisolasi dari rangkaian kumparan :

- Normal terbuka [*normally opened*] : kontak-kontak tertutup bila relay dienergikan.
- Normal tertutup [*normally closed*] : kontak-kontak terbuka bila relay dienergikan.
- Tukar-sambung [*changeover*] : relay ini mempunyai kontak tengah yang normalnya tertutup tetapi "melepaskan" diri dari posisi ini dan membuat kontak dengan yang lain bila relay dienergikan.

Berikut ini beberapa jenis relay :

1. Relay Lidi

   Pada relay ini hanya mempunyai satu kontak yang berbeda pada tabung gelas yang dililit kumparan. Relay ini sangat peka, hanya dengan arus yang kecil (10…30 mA bergantung dari banyaknya lilitan kumparan) sudah cukup untuk memberikan tenaga pada kumparannya.

2. Relay konvensional

   pada relay ini terdiri atas beberapa mekanik yang dapat bergerak, jika kumparan dialirkan arus listrik. Relay ini membutuhkan arus yang lebih besar daripada relay lidi. Arus yang dibutuhkan oleh relay ini berkisar 500mA. Relay ini memiliki beberapa sistem hubungan kontak arus, yaitu SPST, SPDT dan DPDT.

   SPST (*Single Pole Single Throw*) yaitu relay yang mempunyai sebuah masukan dan sebuah keluaran. SPDT (*Single Pole Double Throw*) yaitu relay yang memiliki satu masukan dan dua keluaran. DPDT (*Double Pole Double Throw*) memiliki dua masukkan dan masing-masing mempunyai dua keluaran.

D. LCD (Liquid Crystal Display)

LCD merupakan suatu komponen *opto electronic* yang berfungsi sebagai alat penampil elektronik yang mirip dengan monitor dan diaktifkan dengan molekul kristal cair yang merupakan unsur utamanya.

Berdasarkan tampilannya LCD dapat dikelompokkan menjadi beberapa jenis, diantaranya seperti :

1. *Segmented* LCD, jenis ini sering digunakan untuk jam digital, dan alat ukur digital.
2. *Dot Matrix Character* LCD, jenis ini memiliki beberapa kombinasi dalam menentukan jumlah karakter yang ditampilkan oleh LCD. Contohnya seperti 2 baris x 20 karakter atau 4 baris x 20 karakter.

Register yang terdapat dalam LCD adalah sebagai berikut :

a. IR (*Instruction Register*), untuk menentukan fungsi yang harus dikerjakan LCD serta untuk pengalamatan DD-RAM atau CG-RAM.
b. DR (*Data Register*), sebagai tempat data DD-RAM atau CG-RAM yang akan dituliskan ke atau dibaca oleh komputer atau sistem minimum. Saat dibaca DR menyimpan data DD-RAM atau CG-RAM, setelah itu data alamat berikutnya secara otomatis masuk ke DR. Pada waktu penulisan, cukup mengatur awal DD-RAM atau CG-RAM, maka untuk selanjutnya data dituliskan ke DD-RAM atau CG-RAM sejak alamat awal tersebut.
c. BF (*Busy Flag*), untuk memberikan tanda bahwa LCD dalam keadaan siap atau sedang sibuk. Apabila LCD sedang melakukan operasi internal, BF diset menjadi satu, sehingga tidak akan menerima perintah eksternal. Jadi BF harus dicek apakah telah di-*reset* menjadi nol, supaya LCD dapat ditulis , yaitu dengan RS = 0 dan R/W = 1.
d. AC (*Address Counter*), untuk menunjuk alamat DD-RAM atau CG-RAM yang telah diprogram dengan address command. Setelah DD-RAM atau CG-RAM dibaca atau ditulis, maka AC secara otomatis menunjuk alamat berikutnya. Alamat yang disimpan AC dapat dibaca bersamaan dengan BF.
e. DD-RAM (*Display Data Random Access Memory*), sebagai data *storage* sebesar 80 byte. AC menunjuk alamat karakter yang sedang ditampilkan. Berikut posisi tampilan dan alamat relatifnya setiap DD-RAM.

Alamat DD-RAM pada LCD 40 x 2

Tabel 2.4 LCD 40 x 2 karakter

Karakter ke	1	2	3	4	5	6	7	8	9	…	39	40
Baris ke-1	00h	01h	02h	03h	04h	05h	06h	07h	08h	…	26h	27h
Baris ke-2	40h	41h	42h	43h	44h	45h	46h	47h	48h	…	66h	67h

Tabel 2.5 LCD 20 x 4 karakter

Karakter ke	1	2	3	4	5	6	7	8	9	…	19	20
Baris ke-1	00h	01h	02h	03h	04h	05h	06h	07h	08h	…	26h	27h
Baris ke-2	40h	41h	42h	43h	44h	45h	46h	47h	48h	…	52h	53h
Baris ke-3	14h	15h	16h	17h	18h	19h	1Ah	1Bh	1Ch	…	26h	27h
Baris ke-4	54h	55h	56h	57h	58h	59h	5Ah	5Bh	5Ch	…	66h	67h

Bila terjadi *shift* (geser), maka data yang terdapat pada posisi paling kiri/kanan tidak hilang, melainkan berpindah ke posisi paling kanan kiri. Jadi data tampilan tetap sama, hanya posisinya yang bergeser.

f. CG-ROM (*Character Generator Read Only Memory*), pada LCD sudah terdapat ROM untuk menyimpan karakter-karakter ASCII (*American Standar Code for Interchange Instruction*), sehingga cukup untuk memasukkan kode ASCII untuk menampilkannya. Tampilan karakter pada LCD akan dilampirkan.

g. CG-RAM (*Character Generator Random Access Memory*), sebagai data *storage* untuk merancang karakter yang dikehendaki. Untuk CG-RAM terletak pada

kode ASCII dari 00h sampai 0Fh, tetapi hanya delapan karakter disediakan. Alamat CG-RAM hanya 6 bit, 3 bit tinggi menjadi 3 bit rendah DD-RAM yang menunjukkan kode karakter, sedangkan 3 bit rendah sebagai posisi data CG-RAM untuk membuat tampilan satu baris dalam *dot matriks* 5 x 7 karakter tersebut, dimulai dari atas. Sehingga karakter untuk kode ASCII 00h sama dengan 08h, 01h dengan 09h sampai 07h dengan 0Fh. Dengan demikian untuk perancangan satu karakter memerlukan penulisan data ke CG-RAM sampai 8 kali.

h. *Cursor and Blink Control Circuit* merupakan rangkaian yang menghasilkan tampilan kursor dan kondisi *blink*.

Insruksi yang dapat diberikan ke modul LCD :

- *Display Clear* : Untuk mereset AC (*Address Counter*) menjadi 00h sehingga kursor akan menunjuk ke posisi paling awal dari *display*, dan semua DD-RAM berisi karakter *space* (20h) akibatnya layar LCD dalam kondisi kosong (tidak ada karakter apapun).

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	0

- *Display Cursor Home* : Untuk mereset AC ke 00h, sehingga kursor berada di posisi awal *display*. Disini isi dari DD-RAM tidak berubah.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	X

X : *don't care*.

- *Entry Mode Set* : sebagai penentu *increament / decreament* dari nilai AC. Bila I/D = 1, maka AC akan bertambah 1 setelah baca/tulis ke DD-RAM atau CG-RAM, sehinggan akibatnya kursor bergeser ke kanan. Jika I/D = 0 yang terjadi adalah kebalikannya. Sedangkan apabila S = 1, maka semua data akan digeser satu posisi ke kanan (I/D = 1) atau ke kiri (I/D = 0) setelah melakukkan penulisan ke DD-RAM. Bila S = 0, maka tidak ada pergesaran. Hal ini tidak berlaku jika melakukan penulisan ke CG-RAM.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	I/D	S

- *Display On Off* : Untuk menentukan mode dari *display cursor*.

  Jika D = 1, maka *display* LCD aktif (*ON*) dan data pada DD-RAM ditampilkan ke layar. Sedangkan jika D = 0 , yang terjadi adalah *display* LCD akan *OFF*

  Jika C = 1, maka kursor *ON*, begitu juga sebaliknya.

  Jika B = 1, maka maka kursor akan berkedip (*blinking*), dan begitu juga sebaliknya.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	D	C	B

- *Display Cursor Shift* : untuk menentukan pergesaran kursor.

  Jika S/C = 1, maka kursor berpindah satu layar penuh (*shift display*) ke kiri atau kanan.

Jika S/C = 0, maka kursor berpindah satu posisi ke kiri atau kanan.

Jika R/L = 1 bergeser ke kanan, dan jika R/L = 0 bergeser ke kiri.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	S/C	R/L	X	X

- *Function Set* : untuk menentukan jenis data apakah 8 bit atau 4 bit, dan digunakan untuk memilih modul LCD.

  Jika D1 = 1, maka 8 bit.

  Jika D1 = 0, maka 4 bit.

  Jika N = 1, maka modul LCD 40x2.

  Jika N = 0, maka modul LCD 20x4.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	D1	N	0	X	X

- CG-RAM *address Set* : Digunakan untuk menentukan alamat CG-RAM, sehingga dapat dilakukan penulisan ke dalam CG-RAM ataupun membaca isi dari CG-RAM. Variabel A menunjukkan alamat dari CG-RAM yang dimulai dari DB0…DB5.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	A	A	A	A	A	A

- DD-RAM *address Set* : untuk menentukan alamat awal (*starting address*) dari karakter yang ingin ditampilkan. Variabel A merupakan alamat dari DD-RAM dimulai dari DB0…DB6.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	A	A	A	A	A	A	A

- *Busy Flag / Address Counter Read* : untuk membaca status dari LCD.

  Jika BF = 1, maka modul LCD sedang sibuk dan tidak dapat menerima instruksi lain. Jika BF = 0 yang terjadi adalah sebaliknya.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	BF	A	A	A	A	A	A	A

- CG-RAM / DD-RAM *Data Write* : untuk menulis karakter di alamat yang ditunjuk AC (*Address Counter*).

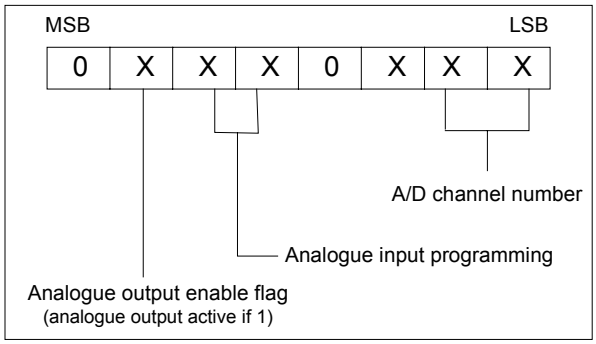
RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D	D	D	D	D	D	D	D

- CG-RAM / DD-RAM *Data Read* : untuk membaca karakter di alamat yang ditunjuk AC.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	D	D	D	D	D	D	D	D

2.5 Sinyal Konversi

Sinyal yang diterima oleh sistem harus terlebih dahulu dikonversikan ke bentuk yang dapat dimengerti oleh aktuator. Oleh karena itu akan digunakan ADC dan DAC. ADC (*Analog to Digital Converter*) adalah digunakan untuk mengubah dari sinyal analog menjadi digital, misalnya proses pengkonversian dari hasil pengukuran suhu air yang kemudian ditampilkan ke dalam LCD. DAC (*Digital to Analog Converter*) adalah digunakan untuk mengubah dari sinyal digital menjadi analog, misalnya proses pengkonversian input suhu yang diberikan oleh *user* yang ditampilkan di LCD kemudian diubah ke bentuk yang dibutuhkan oleh *heater* untuk mengendalikan *heater*. Pada sistem ini digunakan DT51 I²C ADDA yang dapat digunakan untuk mengubah sinyal analog seperti tegangan atau arus ke data biner dan sebaliknya. Pada DT51 I²C terdapat empat channel 8 bit analog input (*Analog to Digital Converter*/ADC), dan satu channel 8 bit analog output (*Digital to Analog Converter*/DAC). Apakah DT51 I²C ADDA ini akan berfungsi sebagai ADC ataupun DAC ditentukan dari control byte-nya, lihat gambar 2.12. Untuk fungsi ADC dapat diprogram ke dalam bentuk single ended input ataupun differential input, hal ini berdasarkan nilai bit yang diberikan. Dalam sistem ini yang digunakan adalah single ended maka bit yang digunakan adalah 00.



Gambar 2.12 Control Byte

Untuk aturan dalam penggunaan fungsi ADC maupun DAC adalah sebagai berikut :

1. Fungsi ADC

   Input : Accumulator A

   Output : Variabel Ch0-Ch3

      Keterangan :

      Register accumulator A (0-7) : untuk memilih satu dari 8 alamat board DT51 I2C ADDA yang ada.

      Ch0 : adalah variabel tempat disimpannya hasil ADC pada analog input 0

      Ch1 : adalah variabel tempat disimpannya hasil ADC pada analog input 1

      Ch2 : adalah variabel tempat disimpannya hasil ADC pada analog input 2

      Ch3 : adalah variabel tempat disimpannya hasil ADC pada analog input 3

2. Fungsi DAC

   Input : Accumulator A dan register B

   Output : Flag OuputEnb

      Keterangan :

      Register B : diisi dengan data digital yang akan dikonversi menjadi tegangan analog pada pin AOUT.

      Flag OutputEnb (“1/0”) : jika set (OuputEnb=”1”) maka hasil konversi DAC dioutputkan pada pin AOUT, jika reset (OutputEnb=”0”) maka hasil konversi DAC tidak dioutputkan pada pin AOUT, Flag ini otomatis set jika rutin WriteDAC dijalankan.

Setiap board dari DT51 I²C ADDA dilengkapi jumper untuk setting alamat, sehingga dapat diekspan sampai dengan 8 *board* tanpa tambahan perangkat keras.

2.6 System bus I²C

Umumya pada saat ini didalam desain, elektronik dituntut untuk semakin ringkas dan fleksibel, dimana ukuran fisik IC semakin diperkecil dan jumlah pin diminimalkan dengan tetap menjaga fleksibilitas dan kompabilitas IC sehingga mudah untuk digunakan dalam berbagai keperluan desain yang berbeda. Dengan pengunaan sistem I²C, hal ini memungkinkan dimana sistem I²C ini menggunakan konsep komunikasi dua arah antar IC atau antar sistem secara serial dengan hanya mengunakan 2 jalur, yaitu SCL (*serial clock*) dan SDA (*serial data*). Peralatan yang terhubung pada jalur I²C dapat dioperasikan sebagai *Master* atau *Slave*.

Peralatan *master* mengontrol jalur komunikasi bus dengan inisialisasi atau pemutusan transfer data, dan membangkitkan *clock* untuk sistem I²C. sementara peralatan *slave* menunggu untuk diamati dan dikontrol oleh *master*. Dalam suatu waktu *slave* dapat berfungsi sebagai pengirim data dan *master* sebagai penerimanya. Sebagai contoh jika *master* meminta data dari *slave*, maka permintaan data ini akan diterima oleh *master* dari *slave*. Dalam keadaan ini pembangkitan *clock* tetap dilakukan oleh *master*.

Beberapa definisi dari istilah pada I²C bus :

Transmitter : peralatan yang mengirim data ke bus.

Receiver : peralatan yang menerima data dari bus.

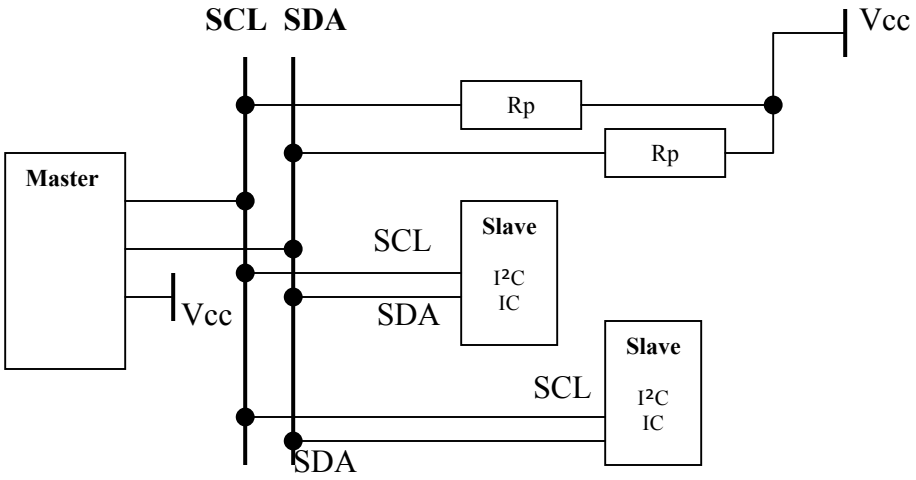
Master : peralatan yang memiliki inisiatif (memulai dan mengakhiri) transfer data dan yang membangkitkan sinyal clock.

Slave : peralatan yang dialamati (diakses berdasarkan alamatnya) oleh master.

Multi-master : yaitu sistem yang memungkinkan lebih dari satu master melakukan inisiatif transfer data dalam waktu yang bersamaan.

2.6.1 Karakteristik hardware I²C

Jalur SCL dan SDA terhubung dengan tegangan positif melalui resistor pull-up pada masing-masing jalur. Gambar 2.13 dibawah ini memperlihatkan konfigurasi I²C-bus secara umum.



Gambar 2.13 System I²C Bus Master/Slave

Sebagai hasil dari adanya resistor pull-up kedua jalur akan berlogika *high* ketika jalur bus tidak dipakai.

2.6.2 Karakteristik protocol I²C

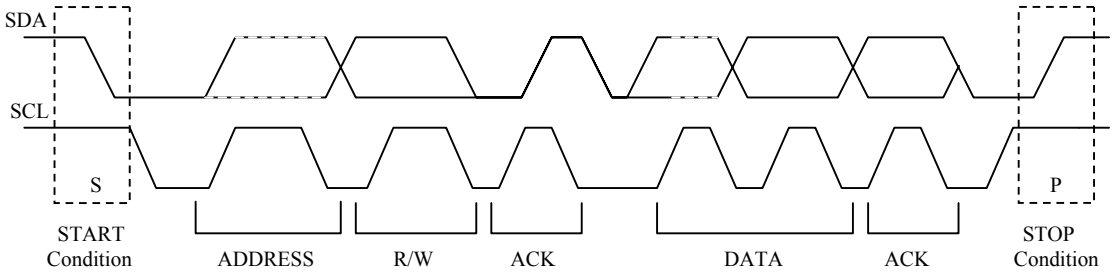
Untuk gambar frame transfer data I²C protocol dapat dilihat pada gambar 2.14.

S	Slave Address	R/W	Data	A	Data	A/Ā	P

S : START condition (Master)
A : ACKnowledge (Slave)
R/W : Read/Write (Master)
P : STOP condition (Master)
Ā : Not ACKnowledge (Slave)

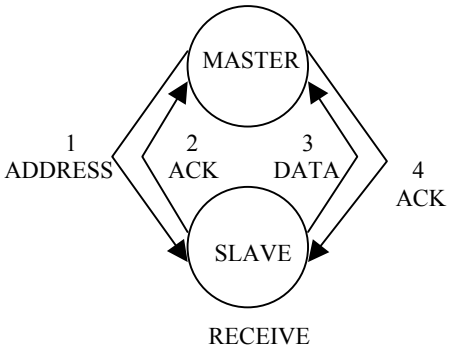
Gambar 2.14 Frame Transfer Data I²C Protocol

Bagian terpenting dari frame pada gambar 2.14 adalah kondisi START / STOP, alamat slave, dan data dengan ACK. Struktur frame data transfer tetap sama kecuali jumlah data yang dikirim dan arah pengiriman. Semua fungsi dari frame tersebut dibangkitkan oleh master kecuali ACK. Gambar 2.15 menampilkan lebih jelas diagram waktu, urutan dari sinyal control dan pengiriman data.



Gambar 2.15 Data Transfer I^2C Secara Lengkap

2.6.3 Proses Receive

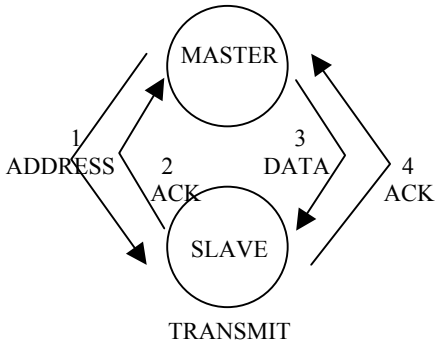


Gambar 2.16 Diagram State Receive

Cara *Master* menerima data sebagai berikut :

- Proses 1, *Master* mengirim alamat *Slave*.
- Proses 2, jika *Slave* yang dialamati ada maka *Slave* tersebut mengirim sinyal ACK ke *Master*.
- Proses 3, *Slave* mengirim data ke *Master*.
- Proses 4, *Master* mengirim ACK ke *Slave*, jika *master* telah berhasil menerima data. Jika *Master* sudah mendapatkan semua data, maka *Master* akan mengirim NACK diikuti dengan sinyal STOP.

2.6.4 Proses Transmit



Gambar 2.17 Diagram State Transmit

Cara *Master* mengirim data sebagai berikut :

- Proses 1, *Master* mengirim alamat *Slave*.
- Proses 2, jika *Slave* yang dituju ada maka akan dikirim sinyal ACK ke *Master*.
- Proses 3, *Master* mengirim data ke *Slave*.
- Proses 4, *Slave* akan mengirim sinyal ACK ke *Master* jika data sudah diterima oleh *Slave*.