

2. TEORI PENUNJANG

2.1. Kompas

Kompas adalah alat penunjuk arah yang digunakan untuk mengetahui arah utara magnetis. Arah utara magnetis ini berbeda dengan arah utara sebenarnya. Arah utara sebenarnya adalah arah utara yang mengarah pada kutub utara bumi sedangkan arah utara magnetis ini tidak tepat berada di kutub utara bumi, akan tetapi berada di Jazirah Boothia, di utara Kanada. Karena sifat kemagnetannya, jarum kompas akan menunjuk arah utara-selatan (jika tidak dipengaruhi oleh adanya gaya-gaya magnet lainnya selain magnet bumi).

Secara fisik, kompas terdiri atas:

- Badan, yaitu tempat komponen-komponen kompas lainnya berada.
- Jarum, selalu mengarah utara selatan bila kompas diletakan secara horisontal.
- Skala penunjuk, menunjukkan derajat sistem mata angin.

Ada beberapa jenis-jenis kompas, pada umumnya dipakai dua jenis kompas, yaitu kompas bidik (misalnya kompas prisma) dan kompas *orienteeering* (misalnya kompas silva). Kompas bidik mudah digunakan untuk membidik, tetapi dalam pembacaan di peta perlu dilengkapi dengan busur derajat dan penggaris. Kompas silva kurang akurat (jika dipakai untuk membidik), tetapi banyak membantu dalam pembacaan dan perhitungan di peta.

Secara prinsip, tidak ada perbedaan yang mencolok dalam tiap-tiap tipe kompas, yang menjadi perbedaan hanyalah *packing* kompas tersebut yang disesuaikan dengan fungsi dan kegunaan masing-masing kompas. Oleh sebab itu untuk memahami cara penggunaan masing-masing tipe kompas, perlu dipelajari terlebih dahulu.

Kompas yang baik pada ujungnya dilapisi fosfor agar dapat menyala dalam keadaan gelap. Gambar berikut ini adalah contoh dari sebuah kompas *lensatic* (jenis kompas prisma).



Gambar 2.1. Kompas Analog Jenis Kompas Prisma.

Sumber: Vtarmynavy. *Silva Lensatic 360*. 15 Februari 2006
 <<http://www.vtarmynavy.com/silva-lensatic-360.htm>>

Selain kompas, terdapat pula *General Positioning System* (GPS), alat ini memiliki kompas digital dengan ketelitian adalah per satu derajat. Namun kompas GPS ini tidak dapat dipergunakan untuk kegiatan yang berada di dalam ruangan. Hal ini dikarenakan GPS ini menggunakan satelit sebagai sarana informasi dan *database*-nya, sehingga membutuhkan area terbuka (*clear sky*).



Gambar 2.2. Garmin's *General Positioning System* (GPS)

Sumber: Garmin. *Garmin's GPS*. 15 Februari 2006.
 <http://www.gps-y.pl/zdjecia2_60cs.html>

Kompas dipakai dengan posisi horisontal sesuai dengan arah medan magnet bumi. Dalam pemakaian kompas, perlu dijauhkan dari pengaruh benda-benda yang mengandung magnet atau logam, seperti pisau, karabiner, jam tangan dan lainnya. Kehadiran benda-benda tersebut akan mempengaruhi jarum kompas sehingga ketepatannya akan berkurang.

2.2. Mikrokontroler AT89S53

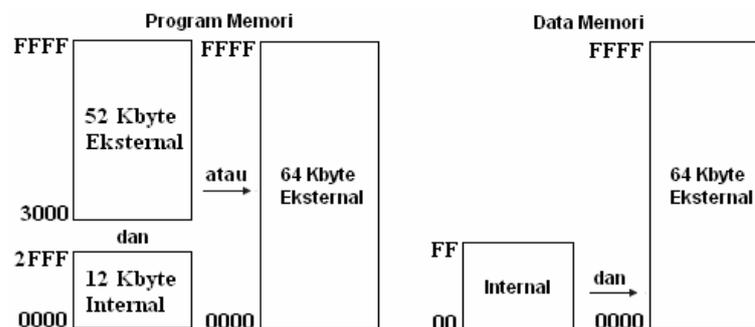
Mikrokontroler adalah sebuah komponen elektronik yang dapat bekerja sesuai dengan program yang diisikan ke dalam memorinya seperti layaknya sebuah komputer yang sederhana. Ada beberapa macam tipe mikrokontroler dengan berbagai macam kemampuan dan ukuran fisik.

Salah satu mikrokontroler yang ada dan yang digunakan dalam tugas akhir ini adalah AT89S53. AT89S53 adalah mikrokontroler dengan arsitektur MCS-51 produksi ATMEL yang mempunyai sistem memori, *timer*, port serial, dan 32 *bit* I/O di dalamnya, sehingga sangat memungkinkan untuk membentuk suatu sistem yang hanya terdiri dari *single chip* (keping tunggal) saja.

AT89S53 memiliki kemampuan sebagai berikut:

- 8-bit CMOS kontroler
- 12Kbytes *flash memory*
- 256 byte *internal* RAM
- 32 jalur *input-output*
- *Three* 16-bit *timer/counters*
- Batasan tegangan operasinya adalah 4 Volt sampai 6 Volt

AT89S53 memiliki ruang alamat terpisah untuk program memori dan data memori. Berikut ini adalah gambar yang menunjukkan peta program memori dan data memori pada AT89S53.



Gambar 2.3. Peta Memori AT89S53

Sumber: Atmel Corporation. *Memory Organization*. 10 Februari 2006. p.21
<<http://www.nalanda.nitc.ac.in/industry/datasheets/atmel/acrobat/doc0498.pdf>>

2.2.1. Deskripsi *Pin* dari AT89S53

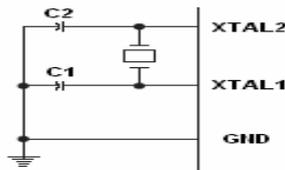
Konfigurasi *pin* dari AT89S53 dapat dilihat pada gambar berikut ini.

(T2) P1.0	1	40	VCC
(T2Ex) P1.1	2	39	P0.0 (AD0)
P1.2	3	38	P0.1 (AD1)
P1.3	4	37	P0.2 (AD2)
(SS) P1.4	5	36	P0.3 (AD3)
(MOS) P1.5	6	35	P0.4 (AD4)
(MISO) P1.6	7	34	P0.5 (AD5)
(SCK) P1.7	8	33	P0.6 (AD6)
RST	9	32	P0.7 (AD7)
(RXD) P3.0	10	31	EA/VPP
(TXD) P3.1	11	30	ALE / PROG
(INT0) P3.2	12	29	PSEN
(INT1) P3.3	13	28	P2.7 (A15)
(T0) P3.4	14	27	P2.6 (A14)
(T1) P3.5	15	26	P2.5 (A13)
(M2) P3.6	16	25	P2.4 (A12)
(RD) P3.7	17	24	P2.3 (A11)
XTAL2	18	23	P2.2 (A10)
XTAL1	19	22	P2.1 (A9)
GND	20	21	P2.0 (A8)

Gambar 2.4. Konfigurasi *Pin* AT89S53

Sumber: Atmel Corporation. *AT89S53 Datasheet*. 10 Februari 2006. p.2
<<http://www.atmel.com/atmel/acrobat/doc0787.pdf>>

1. VCC, *pin* untuk sumber tegangan antara 4 Volt sampai 6 Volt.
2. GND, *pin* untuk *ground*.
3. *Port* 0 (P0.0 sampai P0.7), *port* 0 adalah 8-bit I/O *port* dua arah. *Port* ini dapat dikonfigurasi untuk berfungsi sebagai *multiplexer address/data bus* pada waktu mengakses eksternal data memori dan eksternal program memori. Eksternal *pull-ups* diperlukan selama verifikasi program.
4. *Port* 1, *port* 2, dan *port* 3 (Px.0 sampai Px.7), *port-port* ini adalah 8-bit I/O *port* dua arah dengan internal *pull-ups*.
5. RST, *pin* ini adalah *pin* untuk fungsi *reset*. Sinyal *high* pada *pin* ini selama dua putaran mesin akan menjalankan fungsi *reset* alat ini.
6. XTAL1, *input* untuk *inverting oscillator amplifier* dan *input* untuk mengoperasikan rangkaian *clock* di dalamnya.
7. XTAL2, *output* dari *inverting oscillator amplifier*.



Gambar 2.5. Rangkaian osilator pada AT89S53

Sumber: Atmel Corporation. *AT89S53 Datasheet*. 10 Februari 2006. p.17
<<http://www.atmel.com/atmel/acrobat/doc0787.pdf>>

2.2.2. Operasi *Timer*

AT89S53 memiliki tiga buah *timer*, yaitu Timer 0, Timer 1, dan Timer 2. Ketiga *timer* tersebut dapat berfungsi sebagai *counter* ataupun sebagai *timer*. Perbedaan antara *timer* dan *counter* adalah jika *timer* memiliki sumber *clock* dengan frekuensi tertentu yang sudah pasti, sedangkan *counter* mendapatkan sumber *clock* dari pulsa yang hendak dihitung jumlahnya. Aplikasi dari keduanya juga berbeda, disesuaikan dengan kebutuhan dari program yang dibuat. Untuk *timer*, biasanya digunakan untuk aplikasi menghitung lamanya suatu kejadian yang terjadi. Sedangkan untuk *counter*, digunakan untuk menghitung jumlah kejadian yang terjadi pada suatu periode waktu tertentu.

Keduanya (*timer* dan *counter*) masing-masing memiliki 16-bit *timer* yang mampu diatur keaktifannya maupun mode operasinya serta dapat di-*reset* dan di-*set* dengan harga tertentu. Untuk mengatur *timer/counter* pada Timer 0 dan Timer 1 ini, AT89S53 memiliki empat buah *Special Function Register* (SFR), yaitu TMOD (*Timer Mode Register*), THx dan TLx (16-bit *Register*) serta TCON (*Timer Control Register*). Sedangkan untuk mengatur *timer/counter* pada Timer 2, AT89S53 memiliki empat buah *Special Function Register* (SFR), yaitu T2CON (*Timer/Counter 2 Control Register*), T2MOD (*Timer 2 Mode Control Register*), TH2 dan TL2 (16-bit *Register*).

2.2.2.1. *Timer Mode Register* (TMOD)

Timer 1				Timer 0			
Gate	C/T	MO	M1	Gate	C/T	MO	M1
{1}	{1}	{1}	{1}	{0}	{0}	{0}	{1}

Gambar 2.6. *Register* TMOD

Register TMOD berupa 8-bit register yang tidak dapat diakses per-bit (*not bit addressable*) yang terletak pada alamat 88_H dengan fungsi-fungsi setiap bit-nya adalah sebagai berikut:

- Gate: *Timer* akan berjalan jika bit ini di-set dan port INT0 (untuk Timer 0) atau port INT1 (untuk Timer 1) berkondisi *high*.
- C/T: Bila bit ini di-set, maka akan berfungsi sebagai *counter*, dan apabila bit ini di-clear, maka akan berfungsi sebagai *timer*.
- M1 dan M0: Untuk memilih *mode* yang digunakan.

2.2.2.2. THx dan TLx

Timer 0 dan *timer* 1 pada AT89S53 terdiri atas 16-bit timer. Masing-masing tersimpan dalam dua buah register, yaitu THx untuk *timer high byte* dan TLx untuk *timer low byte*.

- TH1: *Timer* 1 *high byte* terletak pada alamat 8D_H
- TL1: *Timer* 1 *low byte* terletak pada alamat 8B_H
- TH0: *Timer* 0 *high byte* terletak pada alamat 8C_H
- TL0: *Timer* 0 *low byte* terletak pada alamat 8A_H

2.2.2.3. *Timer Control Register* (TCON)

Register ini hanya memiliki 4-bit saja dari 8-bit, yang memiliki fungsi yang berhubungan dengan *timer*, yaitu TCON.4 (TR0), TCON.5 (TF0), TCON.6 (TR1), dan TCON.7 (TF1). *Register* TCON ini bersifat *bit addressable* atau dapat diakses per-bit. Fungsi-fungsi dari 4-bit register TCON adalah sebagai berikut:

- TF1: *Timer* 1 *overflow flag* yang akan di-set jika *Timer* 1 *overflow*. Bit ini dapat di-clear oleh *software* atau oleh *hardware* pada saat program menuju ke alamat yang ditunjuk oleh vektor interupsi.
- TR1: *Timer* 1 akan aktif jika bit ini di-set dan apabila bit ini di-clear, maka akan me-non-aktifkan *Timer* 1.
- TF0: *Timer* 0 *overflow flag* yang akan di-set jika *Timer* 0 *overflow*. Bit ini dapat di-clear oleh *software* atau *hardware* pada saat program menuju ke alamat yang ditunjuk oleh vektor interupsi.

- TR0: Timer 0 akan aktif jika *bit* ini di-*set* dan apabila *bit* ini di-*clear*, maka akan me-non-aktifkan Timer 0.

2.2.2.4. Mode Pada Timer 0 dan Timer 1

Timer 0 dan *timer 1* pada AT89S53 memiliki 4 buah *mode* kerja, di mana setiap *mode* memiliki fungsi masing-masing. Penentuan *mode* kerja dari *timer* dilakukan dengan melakukan inisialisasi pada *register* TMOD.

- Mode 0: Pada *mode* ini, *timer* bekerja dengan *mode* 13-bit *timer*
- Mode 1: Pada *mode* ini, *timer* bekerja dengan 16-bit *timer*.
- Mode 2: Pada *mode* ini, *timer* bekerja dengan *mode* 8-bit *timer*.
- Mode 3: Pada *mode* ini, *timer* bekerja dengan memiliki 3 buah *timer*, yaitu *timer 1* sebagai 16-bit *timer*, sedangkan *timer 0* terpisah menjadi dua buah *timer*, yaitu TL0 dengan TF0 sebagai *timer overflow flag*-nya dan TH0 dengan TF1 sebagai *timer overflow flag*-nya.

2.2.2.5. T2CON (*Timer/Counter 2 Control Register*)

Register ini terdiri dari 8-bit yang memiliki fungsi yang berhubungan dengan *timer*, yaitu TF2 (*bit 7*), EXF2, (*bit 6*), RCLK (*bit 5*), TCLK (*bit 4*), EXEN2 (*bit 3*), TR2 (*bit 2*), C/ $\overline{T2}$ (*bit 1*), CP/ $\overline{RL2}$ (*bit 0*). Fungsi-fungsi dari 8-bit *register* T2CON tersebut adalah sebagai berikut:

- TF2: Timer 2 *overflow flag* yang akan di-*set* bila Timer 2 *overflow*. *Bit* ini harus di-*clear* oleh *software*.
- EXF2: Timer 2 *external flag* yang di-*set* pada *mode capture* atau *reload*, yang disebabkan oleh transisi negatif dari T2EX dan EXEN2 = 1.
- RCLK: *Receive Clock Enable*. Bila RCLK = 1, maka menyebabkan pulsa dari Timer 2 *overflow* digunakan untuk menerima *clock* pada *serial port* Mode 1 dan 3. Jika RCLK = 0, maka Timer 1 *overflow* digunakan untuk menerima *clock*.
- TCLK: *Transmit Clock Enable*. Jika TCLK = 1, maka menyebabkan pulsa dari Timer 2 *overflow* digunakan untuk mengirim *clock* pada

serial port Mode 1 dan 3. Jika $TCLK = 0$, maka Timer 1 *overflow* yang digunakan untuk mengirim *clock*.

- EXEN2: Timer 2 *External Enable*. Jika EXEN2 = 1, maka terjadi mode *capture* atau *reload* sebagai hasil dari transisi negatif pada T2EX (bila Timer 2 tidak digunakan sebagai *clock* pada *serial port*). Jika EXEN2 = 0, maka menyebabkan Timer 2 mengabaikan T2EX.
- TR2: *Bit* yang digunakan untuk mengontrol *start/stop* Timer 2. Jika TR2 = 1, maka Timer 2 akan berjalan (*start*).
- $C/\overline{T2}$: *Bit* untuk memilih Timer 2 sebagai *timer* atau *counter*. Jika $C/\overline{T2} = 0$, maka Timer 2 berfungsi sebagai *timer*. Sebaliknya, jika $C/\overline{T2} = 1$, maka Timer 2 berfungsi sebagai *counter*.
- $CP/\overline{RL2}$: *Bit* untuk memilih mode *Capture* atau *Reload*. Ketika $CP/\overline{RL2} = 1$, maka terjadi mode *capture* pada saat transisi negatif pada T2EX jika EXEN2 = 1. Saat $CP/\overline{RL2} = 0$, maka terjadi *reload* ketika Timer 2 *overflow* atau ketika transisi negatif pada T2EX saat EXEN2 = 1. Ketika RCLK atau TCLK sama dengan satu, *bit* $CP/\overline{RL2}$ ini diabaikan dan *timer* menggunakan *auto-reload* pada Timer 2 *overflow*.

2.2.2.6. T2MOD (Timer 2 Mode Control Register)

Register T2MOD berupa 8-bit register yang tidak dapat diakses per-bit (*not bit addressable*) yang terletak pada alamat 0C9_H dengan fungsi-fungsi setiap bit-nya adalah sebagai berikut:

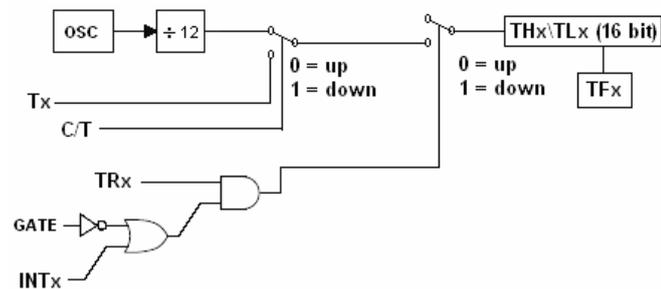
- T2OE: Timer 2 *output enable bit*.
- DCEN: Jika DCEN = 1, maka Timer 2 dikonfigurasi sebagai *up/down counter*.

2.2.2.7. TH2 dan TL2

Timer 2 pada AT89S53 terdiri atas 16-bit timer, yang tersimpan dalam dua buah register, yaitu TH2 untuk *timer 2 high byte*-nya dan TL2 untuk *timer 2 low byte*-nya.

- TH2: Timer 2 *high byte* terletak pada alamat 0CD_H.
- TL2: Timer 2 *low byte* terletak pada alamat 0CC_H.

2.2.2.8. Cara Kerja Timer 0 dan Timer 1



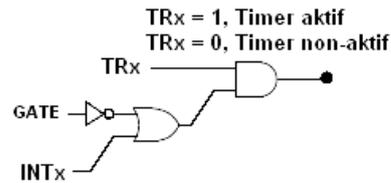
Gambar 2.7. Operasi *Timer*

Sumber: Nawan, Paulus Andi. *Panduan Praktis Teknik Antarmuka dan Pemrograman Mikrokontroler AT89C51*. Jakarta: PT. Elex Media Komputindo, 2003, p.36

Cara kerja dari Timer 0 dan Timer 1 AT89S53 sama seperti pada Timer 0 dan Timer 1 pada AT89C51. Cara kerja dari *timer* ini memerlukan sumber *clock* yang didapat dari eksternal atau internal. Jika *timer* menggunakan sumber *clock* dari eksternal, maka pin T0 (Port3.4) berfungsi sebagai input *clock*. Untuk menjadikan sumber *clock* eksternal sebagai sumber *clock timer*, maka bit C/T dari *register* TMOD harus di-*set* atau berkondisi *high*. Seperti yang tampak pada gambar, jika bit C/T berkondisi *high*, maka saklar akan menghubungkan sumber *clock timer* ke pin Tx (T0 untuk Timer 0 dan T1 untuk Timer 1)

Apabila digunakan sumber *clock* internal, maka *input clock* tersebut berasal dari osilator yang telah dibagi 12. Untuk ini bit C/T dari *register* TMOD harus di-*clear* atau berkondisi *low*, sehingga saklar akan menghubungkan sumber *clock timer* ke osilator yang telah dibagi 12.

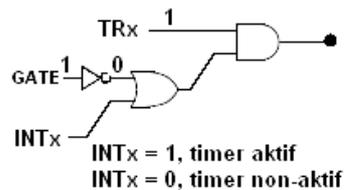
Sedangkan untuk mengaktifkan *timer*-nya dapat dilakukan dengan melalui *software* atau *hardware*. Untuk pengaturan dengan *software*, maka keaktifan *timer* ini ditentukan oleh kondisi bit TRx. Oleh karena itu, *output* dari gerbang OR (yang terhubung ke *input* yang lain dari gerbang AND) harus berlogika *high*. Dengan demikian, jika TRx berlogika *high*, *output* dari gerbang AND akan berlogika *high* dan *timer* akan *aktif*. Dan sebaliknya, apabila TRx berlogika *low*, *output* gerbang AND akan berlogika *low* dan *timer* akan berhenti.



Gambar 2.8. Pengaturan *Timer* dengan *Software*

Sumber: Nawan, Paulus Andi. *Panduan Praktis Teknik Antarmuka dan Pemrograman Mikrokontroler AT89C51*. Jakarta: PT. Elex Media Komputindo, 2003, p.38

Untuk pengaturan *timer* dengan *hardware*, pin INTx berfungsi sebagai penentu. Oleh karena itu, *bit* TRx harus berlogika *high*, agar pengaturan *timer* ditentukan oleh *output* dari gerbang OR. Agar kondisi ini terpenuhi, maka *bit* Gate harus berkondisi *high*. Hal ini sesuai dengan tabel kebenaran yang terdapat pada gambar rangkaian berikut ini.



Gambar 2.9. Pengaturan *Timer* dengan *Hardware*

Sumber: Nawan, Paulus Andi. *Panduan Praktis Teknik Antarmuka dan Pemrograman Mikrokontroler AT89C51*. Jakarta: PT. Elex Media Komputindo, 2003, p.38

Tabel. 2.1. Tabel Kebenaran Pengaturan *Timer* dengan *Hardware*

TRx	GATE	INTx	OUT
1	1	0	0 (<i>Timer Inactive</i>)
1	1	1	1 (<i>Timer Active</i>)

2.2.2.9. Cara Kerja Timer 2

Timer 2 merupakan *timer/counter* 16-bit yang dapat difungsikan sebagai *timer* atau *counter*. Untuk memilih Timer 2 berfungsi sebagai *timer* atau *counter* ini, digunakan *bit* C/T2 pada *Special Function Register* T2CON. Timer 2 memiliki tiga mode operasi, yaitu *capture*, *auto-reload* (*up* atau *down counter*),

dan *baud rate generator*. Mode ini dipilih melalui *bit-bit* pada T2CON. Timer 2 terdiri dari dua *8-bit register*, yaitu TH2 dan TL2.

Sebagai *timer*, *register* TL2 di-*increment* setiap satu siklus mesin (*machine cycle*). Karena satu siklus mesin memiliki osilator selama 12 periode, maka *count rate* adalah sebesar 1/12 dari frekuensi osilator.

Sebagai *counter*, *register* di-*increment* setiap adanya transisi ‘0’ atau ‘1’ dari *pin* input eksternal T2. *Pin input* eksternal tersebut di-*sampling* selama S5P2 tiap siklus mesin. Ketika *sample* menunjukkan kondisi *high* dalam satu siklus dan *low* dalam siklus berikutnya, maka *count* akan di-*increment*. Nilai *count* yang baru muncul pada *register* selama S3P1 dari siklus berikutnya mengikuti adanya transisi yang terdeteksi. Karena dua siklus mesin (24 periode osilator) diperlukan untuk mendeteksi transisi ‘0’ atau ‘1’, maka maksimum *count rate* adalah 1/24 dari frekuensi osilator.

Tabel 2.2. Mode Operasi Timer 2

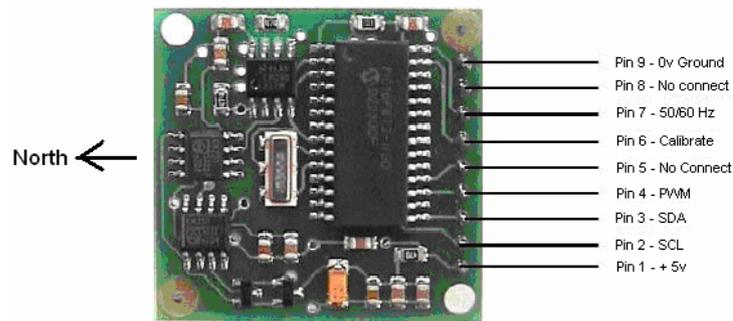
RCLK + TCLK	CP/RL2	TR2	MODE
0	0	1	16-bit Auto-Reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

Sumber: Atmel Corporation. *AT89S53 Datasheet*. 10 Februari 2006. p.9.
<<http://www.atmel.com/atmel/acrobat/doc0787.pdf>>

2.3. Modul Kompas Digital CMPS-03

2.3.1 Cara Kerja Modul Kompas Digital CMPS-03

Modul kompas digital CMPS-03 ini didesain untuk digunakan pada robotika sebagai alat navigasi. Tujuannya adalah untuk menghasilkan suatu besaran tertentu yang dapat diolah untuk kemudian merepresentasikan suatu arah tertentu berdasarkan medan magnet bumi. Gambar berikut ini menunjukkan modul kompas digital CMPS-03 tersebut.



Gambar 2.10. Modul Kompas Digital CMPS-03

Sumber: Robot-Electronics. *CMPS03-Robot Compass Module*. 10 Februari 2006. p.1
 <<http://www.robot-electronics.co.uk/htm/cmeps3doc.shtml>>

Modul ini memerlukan tegangan *supply* sebesar 5 V, dengan arus sebesar 15 mA. Terdapat dua cara untuk mengolah *output* dari modul ini, pertama yaitu *output*-nya berupa sinyal PWM (*Pulse Width Modulation*) pada *pin* nomor 4, atau cara kedua yaitu dengan menggunakan I^2C *interface* pada *pin* 2 dan 3.

Pada tugas akhir ini menggunakan *output* sinyal PWM dari modul kompas digital CMPS-03 ini. Pada *output* PWM ini, yang merepresentasikan besarnya sudut/arah adalah lebar pulsa positif dari sinyal PWM tersebut. Lebar pulsa positifnya bervariasi dari 1 mS (0°) sampai 36,99 mS ($359,9^\circ$). Berarti besar lebar pulsa positifnya bertambah secara linear 100 μ S tiap derajat dengan *offset* +1 mS. Sedangkan lebar pulsa negatifnya adalah 65 mS, dengan demikian satu gelombang PWM besarnya 65 mS ditambah dengan besar lebar pulsa positifnya.

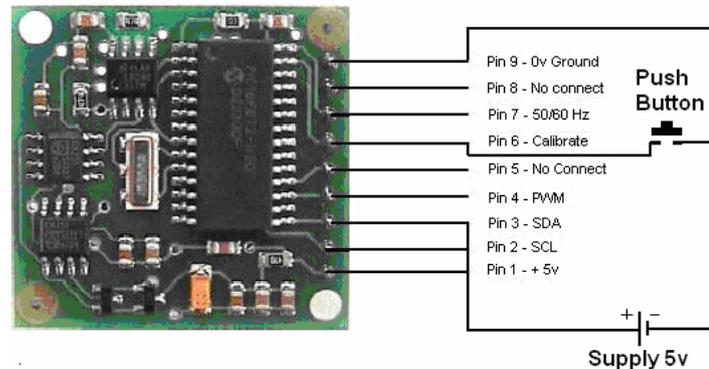
Konfigurasi *pin* dari modul kompas digital CMPS-03 ini adalah sebagai berikut:

1. *Pin* 1, VCC, *pin* untuk sumber tegangan.
2. *Pin* 2 dan *Pin* 3, *pin* untuk aplikasi *output* dengan metode I^2C .
3. *Pin* 4, *pin* untuk aplikasi *output* dengan metode PWM.
4. *Pin* 6, *pin* untuk kalibrasi modul kompas digital CMPS-03.
5. *Pin* 9, *pin* untuk *ground*.

2.3.2. Metode Kalibrasi CMPS-03

Untuk kalibrasi (penyesuaian arah utara modul dengan utara magnetis) terdapat dua metode, yaitu metode I^2C atau dengan metode *pin* 6. Dalam tugas

akhir ini digunakan metode *pin 6*. Hal ini karena metode ini memiliki cara yang sederhana dan sesuai dengan *display* kompas digital yang berupa *stand alone device*. Berikut adalah gambar rangkaian kalibrasi CMPS-03 dengan *pin 6*.



Gambar 2.11. Rangkaian Kalibrasi CMPS-03

Sumber: Wahyudi, Panca. *Pembuatan Display Digital Untuk Modul Kompas CMPS-03*. Surabaya: Universitas Kristen Petra, 2005, p.16

Untuk Proses kalibrasi, *pin 6* diberi sinyal *low* dan kemudian diberi sinyal *high* untuk tiap 4 arah utama kompas, yaitu utara, timur, selatan, dan barat. Proses kalibrasi ini harus searah dengan putaran jarum jam dari utara (0°), timur (90°), selatan (180°), dan barat (270°). Untuk mengetahui arah utara magnetis (0°), digunakan kompas konvensional atau GPS sebagai keperluan kalibrasi ini.

Langkah-langkah kalibrasi:

1. Modul CMPS-03 diletakkan secara mendatar (rata air), mengarah ke utara (0°), kemudian *push button* di-trigger.
2. Modul CMPS-03 diletakkan secara mendatar (rata air), mengarah ke timur (90°), kemudian *push button* di-trigger.
3. Modul CMPS-03 diletakkan secara mendatar (rata air), mengarah ke selatan (180°), kemudian *push button* di-trigger.
4. Modul CMPS-03 diletakkan secara mendatar (rata air), mengarah ke barat (270°), kemudian *push button* di-trigger.

Pin 6 ini memiliki *internal pull-ups* dan setelah proses kalibrasi, maka *pin 6* ini dapat dikondisikan *unconnected*.

2.4. LCD LPH 7779 (48 x 84 Pixels Graphic LCD)

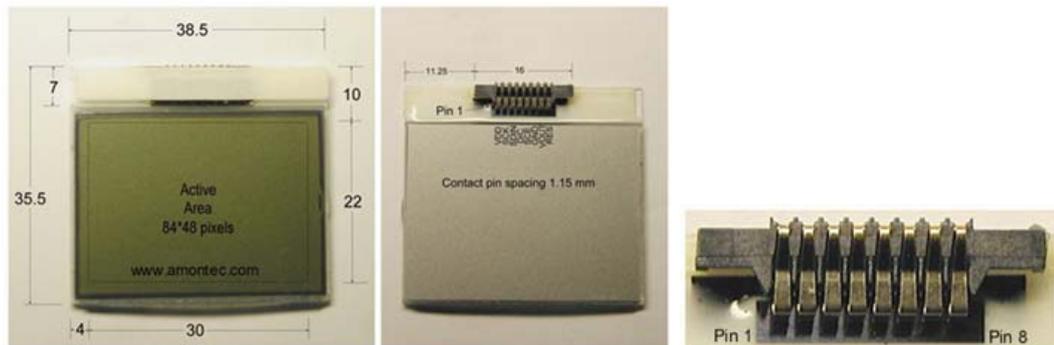
LCD grafik LPH 7779 merupakan sebuah LCD grafik satu warna (*monochrome*) yang memiliki resolusi 48 x 84 *pixel*. LCD ini dapat ditemukan pada beberapa *handphone* produksi Nokia yaitu tipe 3310, 3315, 3330, 3350, dan 3410. Pada LCD ini di dalamnya telah terdapat sebuah IC sebagai *driver/kontroller* LCD 48 x 84 *pixel* yang merupakan produksi Philips yaitu PCD 8544 dengan menggunakan sistem penerimaan data secara serial. Panjang dan lebar dari *display* adalah 38.5 x 35 mm dengan panjang dan lebar *display* yang aktif adalah 30 x 22 mm.

Sebagai komponen eksternalnya perlu ditambahkan sebuah kapasitor dengan nilai minimal sebesar 1 μ F dari Vout ke *ground*. Pemberian kapasitor ini diperlukan agar tampilan pada LCD stabil. Hal ini dapat dibuktikan dengan menampilkan beberapa *pixel* atau karakter pada LCD. Jika tidak diberi sebuah kapasitor dari Vout ke *ground*, tampilan *pixel* atau karakter tersebut akan nampak redup dan beberapa saat kemudian akan hilang. Begitu seterusnya setiap kali diberikan tegangan *supply* pada LCD. Pemberian kapasitor dengan nilai lebih dari 1 μ F dari Vout ke *ground* diperlukan untuk mengurangi *ripple* pada LCD sehingga tampilan pada LCD lebih stabil.

Pada LCD grafik LPH 7779 ini tidak memiliki *on-chip character-set*. Sehingga untuk menampilkan suatu karakter tertentu pada LCD, maka harus dibentuk sendiri melalui beberapa *pixel* individu yang ditampilkan. Dari *pixel-pixel* yang ditampilkan tersebut, maka akan terbentuk karakter yang diinginkan.

2.4.1. Spesifikasi dan Deskripsi *Pin* LCD LPH 7779

Gambar berikut ini menunjukkan spesifikasi mekanik dari LCD grafik LPH 7779.



Gambar 2.12. LCD Grafik LPH 7779

Sumber: Amontec. *Nokia 3310 LCD 84 x 48 pixels*. 15 Februari 2006.
<[http:// www.amontec.com/lcd_nokia_3310.shtml](http://www.amontec.com/lcd_nokia_3310.shtml)>

Tabel 2.3. Keterangan *Pin* LCD LPH 7779

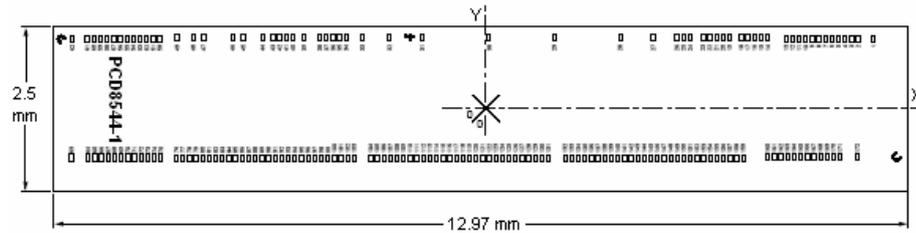
Pin	Sinyal	Deskripsi
1	V_{DD}	<i>Pin</i> untuk tegangan <i>supply logic</i>
2	SCLK	<i>Pin</i> untuk <i>input</i> sinyal <i>clock serial</i>
3	SDIN	<i>Pin</i> untuk <i>input</i> data serial
4	$\overline{D/C}$	<i>Pin</i> untuk <i>Mode Select (Data atau Command)</i>
5	\overline{SCE}	<i>Pin</i> input <i>Chip Enable</i>
6	GND	<i>Pin</i> untuk <i>Ground</i>
7	V_{OUT}	<i>Pin</i> untuk tegangan <i>output</i>
8	\overline{RES}	<i>Pin</i> untuk <i>reset</i> eksternal

Sumber: Amontec. *Nokia 3310 LCD 84 x 48 pixels*. 15 Februari 2006.
<[http:// www.amontec.com/lcd_nokia_3310.shtml](http://www.amontec.com/lcd_nokia_3310.shtml)>

2.5. PCD 8544

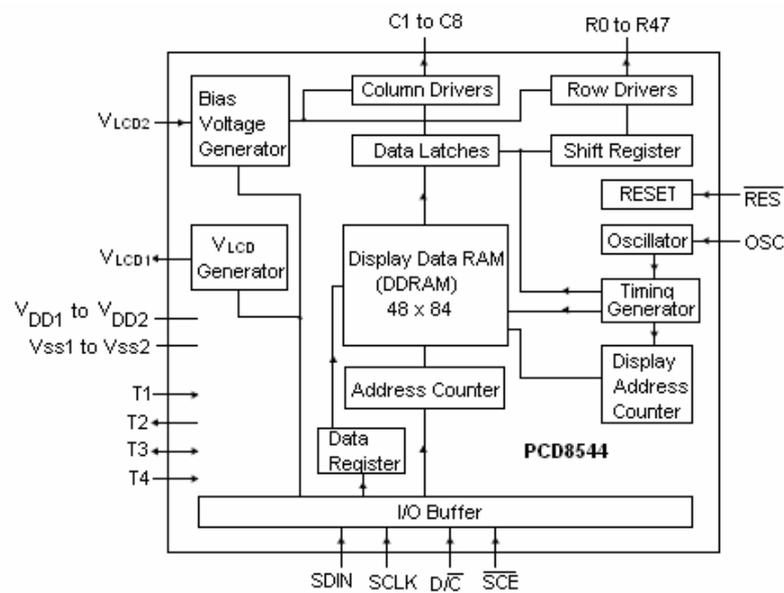
2.5.1. Penjelasan Umum

IC PCD 8544 merupakan sebuah IC *kontroller/driver* dengan teknologi CMOS yang didesain untuk men-*drive* LCD grafik dengan resolusi 48 x 84 *pixel*. PCD 8544 diakses secara serial melalui mikrokontroler dan semua fungsi-fungsi yang diperlukan untuk *display* (LCD) telah tersedia dalam *single chip* ini, termasuk *on-chip generator supply* LCD dan tegangan bias. Aplikasi PCD 8544 digunakan pada peralatan telekomunikasi.



Gambar 2.13. IC PCD 8544

Sumber: Philips Semiconductor. *PCD8544 48 x 84 Pixels Matrix LCD Controller/Driver Datasheet*. 15 Februari 2006. p.24
 <<http://serdisplib.sourceforge.net/ser/doc/pcd8544.pdf>>



Gambar 2.14. Diagram Blok PCD 8544

Sumber: Philips Semiconductor. *PCD8544 48 x 84 Pixels Matrix LCD Controller/Driver Datasheet*. 15 Februari 2006. p.4
 <<http://serdisplib.sourceforge.net/ser/doc/pcd8544.pdf>>

2.5.2. Deskripsi *Pin* PCD 8544Tabel 2.4. Deskripsi *Pin* IC PCD 8544

Simbol	Deskripsi
R0-R47	<i>Output driver</i> baris LCD
C0-C83	<i>Output driver</i> kolom LCD
V _{SS1} , V _{SS2}	<i>ground</i>
V _{DD1} , V _{DD2}	Tegangan <i>supply</i>
V _{LCD1} , V _{LCD2}	Tegangan <i>supply</i> LCD
T1	<i>Input</i> test1
T2	<i>Input</i> test2
T3	<i>Input</i> test3
T4	<i>Input</i> test4
SDIN	<i>Input</i> data serial
SCLK	<i>Input</i> clock serial
D/C	<i>Mode Select (Data atau Command)</i>
SCE	<i>Chip Enable</i>
OSC	Osilator
RES	<i>Input</i> reset eksternal
dummy1, 2, 3, 4	<i>Not-connected</i>

Sumber: Philips Semiconductor. *PCD8544 48 x 84 Pixels Matrix LCD Controller/Driver Datasheet*. 15 Februari 2006. p.5
 <<http://serdisplib.sourceforge.net/ser/doc/pcd8544.pdf>>

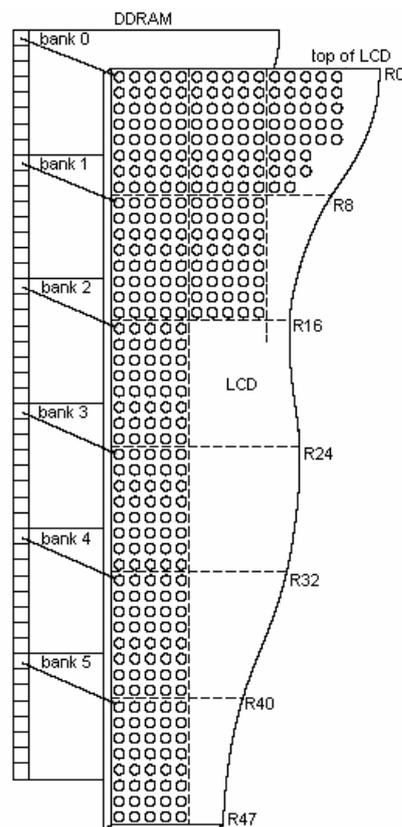
2.5.3. Fitur-Fitur PCD 8544

- Merupakan *single chip LCD controller/driver*
- Memiliki *output* 48 baris dan 84 kolom
- DDRAM 48 x 84 *bit*
- *On-Chip*:
 - Pembangkit tegangan *supply* LCD (Dapat juga di-*supply* secara eksternal)
 - Pembangkit tegangan bias LCD
 - Tidak memerlukan komponen osilator eksternal (Dapat juga diberikan *clock* secara eksternal)
- Memiliki *pin input reset* eksternal (aktif *low*).
- *Serial Interface* maksimal 4.0 Mbits/s
- *Mux rate*: 48

- Tegangan *supply logic* (V_{DD} to V_{SS}) memiliki *range*: 2.7 s/d 3.3 V
- Tegangan *supply LCD* (V_{LCD} to V_{SS}) memiliki *range*:
 - 6.0 s/d 8.5 V bila tegangan LCD dibangkitkan secara internal
 - 6.0 s/d 9.0 V bila tegangan LCD di-*supply* secara eksternal
- Konsumsi daya rendah
- Memiliki temperatur kompensasi dari LCD
- *Temperature range*: -25° s/d $+70^{\circ}\text{C}$

2.5.4. Pengalamatan

Gambar berikut menjelaskan tentang *display mapping* dari DDRAM PCD 8544.



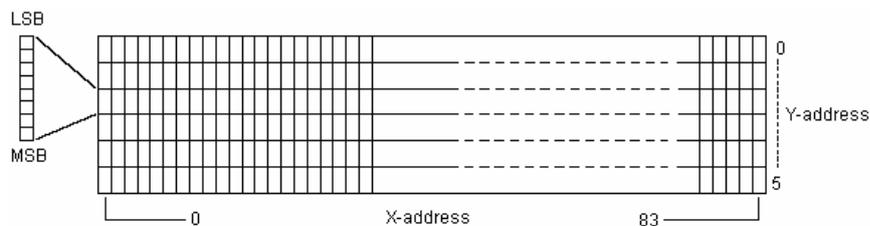
Gambar 2.15. DDRAM to Display Mapping

Sumber: Philips Semiconductor. *PCD8544 48 x 84 Pixels Matrix LCD Controller/Driver Datasheet*. 15 Februari 2006. p.8
 <<http://serdisplib.sourceforge.net/ser/doc/pcd8544.pdf>>

Pada PCD 8544, data dimasukkan ke dalam 48 x 84 *bit* RAM berupa data *byte*. Kolom dialamati dengan menggunakan *address pointer*. *Range* dari alamat

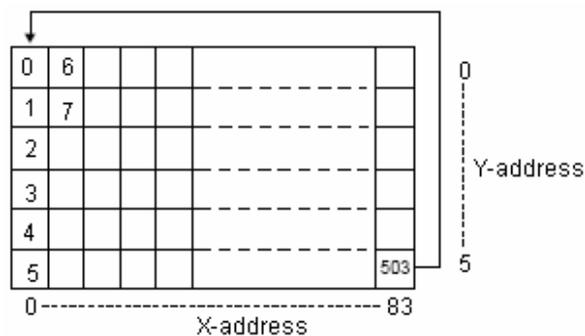
X adalah mulai 0 s/d 83 (1010011) sedangkan alamat Y mulai dari 0 s/d 5 (101). Pengalamatan di luar *range* tersebut tidak dapat dikenali oleh IC PCD 8544.

PCD 8544 memiliki dua mode pengalamatan, yaitu secara horisontal dan vertikal. Pada mode pengalamatan secara vertikal ($V=1$), alamat Y di-*increment* setelah tiap *byte*. Setelah alamat Y yang terakhir ($Y=5$), alamat Y akan kembali lagi mulai dari 0 dan X di-*increment* ke alamat X pada kolom selanjutnya. Pada mode pengalamatan horisontal ($V=0$), alamat X di-*increment* setelah tiap *byte*. Setelah alamat X yang terakhir ($X=83$), alamat X kembali pada 0 dan Y di-*increment* pada alamat ke baris selanjutnya. Setelah alamat terakhir tercapai ($X=83$ dan $Y=5$), *address pointer* kembali lagi pada alamat $X=0$ dan $Y=0$. Gambar berikut menunjukkan format RAM pada PCD 8544 dan mode pengalamatan horisontal dan vertikal.



Gambar 2.16. RAM format, addressing.

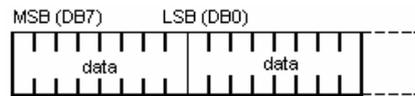
Sumber: Philips Semiconductor. *PCD8544 48 x 84 Pixels Matrix LCD Controller/Driver Datasheet*. 15 Februari 2006. p.9
<<http://serdisplib.sourceforge.net/ser/doc/pcd8544.pdf>>



Gambar 2.17. Sekuen penulisan *byte* data ke dalam RAM dengan pengalamatan vertikal ($V=1$).

Sumber: Philips Semiconductor. *PCD8544 48 x 84 Pixels Matrix LCD Controller/Driver Datasheet*. 15 Februari 2006. p.9
<<http://serdisplib.sourceforge.net/ser/doc/pcd8544.pdf>>

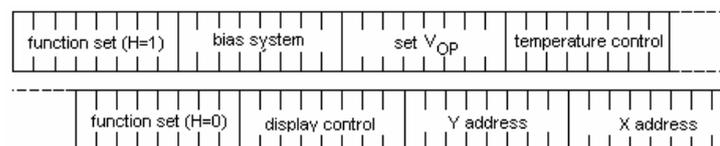
data dan data tersebut disimpan pada DDRAM. Level dari $\overline{D/C}$ dibaca setiap *bit* terakhir dari *byte* data yang dikirimkan. Gambar berikut ini menunjukkan format umum dari aliran data secara serial.



Gambar 2.20. Format umum aliran data serial.

Sumber: Philips Semiconductor. *PCD8544 48 x 84 Pixels Matrix LCD Controller/Driver Datasheet*. 15 Februari 2006. p.11
<<http://serdisplib.sourceforge.net/ser/doc/pcd8544.pdf>>

Pada pengiriman data secara serial, *bit* MSB dikirimkan lebih dulu kemudian baru *bit* selanjutnya sesuai dengan urutannya. Ketika *pin* \overline{SCE} dalam keadaan *high*, sinyal *clock* SCLK diabaikan dan *serial interface* diinisialisasi, tidak ada daya yang dikonsumsi oleh *serial interface* pada kondisi ini. Kemudian pada saat kondisi *low* dari \overline{SCE} , *serial interface* di-*enable*-kan dan hal ini mengindikasikan awal (*start*) dari pengiriman data. Data *serial* kemudian diterima melalui *pin* SDIN dan di-*sampling* setiap kondisi *high* dari sinyal *clock*.



Gambar 2.21. Contoh aliran data serial.

Sumber: Philips Semiconductor. *PCD8544 48 x 84 Pixels Matrix LCD Controller/Driver Datasheet*. 15 Februari 2006. p.11
<<http://serdisplib.sourceforge.net/ser/doc/pcd8544.pdf>>

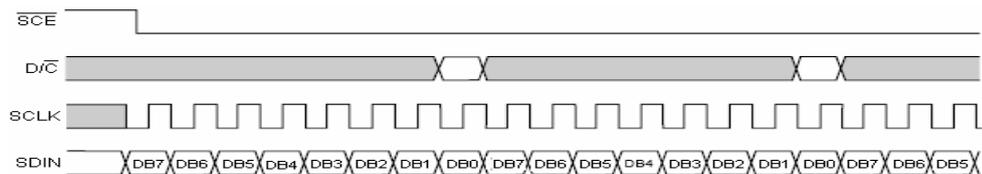
Ketika *pin* \overline{SCE} dalam kondisi *low* setelah *bit* terakhir dari *byte command/data*, *serial interface* akan tetap menunggu *bit* ke-7 pada pengiriman *byte* selanjutnya, pada kondisi *high* dari SCLK.

Bila ada pulsa *reset* pada *pin* \overline{RES} , maka hal ini akan menginterupsi transmisi (tidak ada data yang ditulis pada RAM dan semua *register* berisi nilai '0'). Jika *pin* \overline{SCE} tetap *low* setelah kondisi *high* dari *pin* \overline{RES} , maka *serial interface* siap untuk menerima *bit* ke-7 dari *command/data byte*.



Gambar 2.22. Serial Bus Protocol-transmission of one byte.

Sumber: Philips Semiconductor. *PCD8544 48 x 84 Pixels Matrix LCD Controller/Driver Datasheet*. 15 Februari 2006. p.12
 <<http://serdisplib.sourceforge.net/ser/doc/pcd8544.pdf>>



Gambar 2.23. Serial Bus Protocol- transmission of several byte.

Sumber: Philips Semiconductor. *PCD8544 48 x 84 Pixels Matrix LCD Controller/Driver Datasheet*. 15 Februari 2006. p.12
 <<http://serdisplib.sourceforge.net/ser/doc/pcd8544.pdf>>

Berikut ini adalah tabel yang berisi instruksi yang harus dikirimkan pada IC PCD 8544 sebagai proses inialisasi LCD.

Tabel 2.5. Tabel Instruksi IC PCD 8544

Instruction	D/C	Command Byte								Description
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
(H=0 or 1)										
NOP		0	0	0	0	0	0	0	0	No operation
Function Set	0	0	0	1	0	0	PD	V	H	Power down control; entry mode; extended instruction set control (H)
Write data	1	D7	D6	D5	D4	D3	D2	D1	D0	Write data to DDRAM
(H=0)										
Reserved	0	0	0	0	0	0	1	x	x	Do not use
Display Control	0	0	0	0	0	1	D	0	E	Sets display configuration

Sumber: Philips Semiconductor. *PCD8544 48 x 84 Pixels Matrix LCD Controller/Driver Datasheet*. 15 Februari 2006. p.14
 <<http://serdisplib.sourceforge.net/ser/doc/pcd8544.pdf>>

Tabel 2.6. Tabel Instruksi IC PCD 8544 (lanjutan)

Instruction	D/C	Command Byte								Description
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
Reserved	0	0	0	0	1	x	x	x	x	Do not use
Set Y address of RAM	0	0	1	0	0	0	Y ₂	Y ₁	Y ₀	Set Y addresss of RAM; 0 ≤ Y ≤ 5
Set X address of RAM	0	0	X ₆	X ₅	X ₄	X ₃	X ₂	X ₁	X ₀	Set X addresss of RAM; 0 ≤ X ≤ 83
(H=1)										
Reserved	0	0	0	0	0	0	0	0	1	Do not use
	0	0	0	0	0	0	0	1	x	Do not use
Temperature Control	0	0	0	0	0	0	1	TC ₁	TC ₀	Set temperature coefficient (TC _x)
Reserved	0	0	0	0	0	1	x	x	x	Do not use
Bias System	0	0	0	0	1	0	BS ₂	BS ₁	BS ₀	Set bias system (BS _x)
Reserved	0	0	1	x	x	x	x	x	x	Do not use
Set V _{OP}	0	1	V _{OP6}	V _{OP5}	V _{OP4}	V _{OP3}	V _{OP2}	V _{OP1}	V _{OP0}	Write V _{OP} to register

Sumber: Philips Semiconductor. *PCD8544 48 x 84 Pixels Matrix LCD Controller/Driver Datasheet*. 15 Februari 2006. p.14
<http://serdisplib.sourceforge.net/ser/doc/pcd8544.pdf>

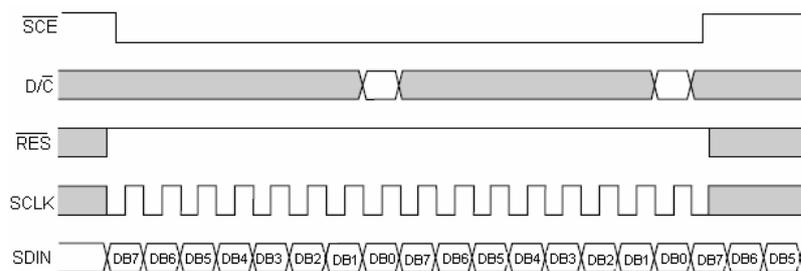
Tabel 2.7. Penjelasan Simbol Pada Tabel Instruksi IC PCD 8544

BIT	0	1
PD	Chip is active	Chip is on power down mode
V	Horizontal addressing	Vertical addressing
H	Use basic instruction set	Use extended instruction set
D and E 00 10 01 11	Display Blank Normal Mode All display segments on Inverse video mode	
TC ₁ and TC ₀ 00 01 10 11	V _{LCD} temperature coefficient 0 V _{LCD} temperature coefficient 1 V _{LCD} temperature coefficient 2 V _{LCD} temperature coefficient 3	

Sumber: Philips Semiconductor. *PCD8544 48 x 84 Pixels Matrix LCD Controller/Driver Datasheet*. 15 Februari 2006. p.14
<http://serdisplib.sourceforge.net/ser/doc/pcd8544.pdf>

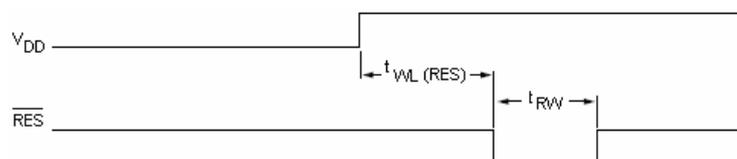
2.5.7. Inisialisasi

Begitu sumber tegangan diberikan, isi dari semua *register* internal dan isi dari RAM dalam kondisi *undefined*, sehingga harus diberikan sinyal *reset* melalui pin $\overline{\text{RES}}$ dengan ketentuan sinyal *reset* harus $\leq 0.3 V_{\text{DD}}$ ketika V_{DD} mencapai V_{DDmin} (atau lebih tinggi) dengan lama waktu maksimum 100 mS (setelah V_{DD} mencapai *high*).



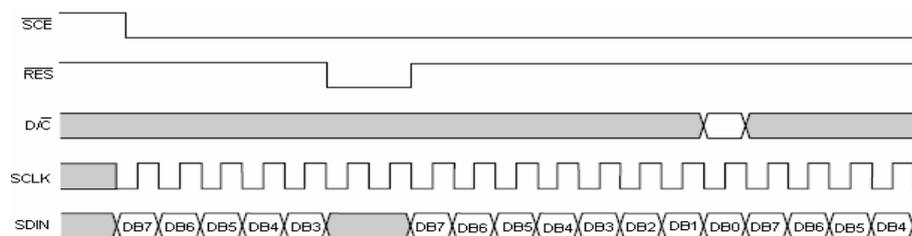
Gambar 2.24. *Serial Bus Reset Function* ($\overline{\text{SCE}}$)

Sumber: Philips Semiconductor. *PCD8544 48 x 84 Pixels Matrix LCD Controller/Driver Datasheet*. 15 Februari 2006. p.13
<<http://serdisplib.sourceforge.net/ser/doc/pcd8544.pdf>>



Gambar 2.25. *Reset Timing*.

Sumber: Philips Semiconductor. *PCD8544 48 x 84 Pixels Matrix LCD Controller/Driver Datasheet*. 15 Februari 2006. p.21
<<http://serdisplib.sourceforge.net/ser/doc/pcd8544.pdf>>



Gambar 2.26. *Serial Bus Reset Function* ($\overline{\text{RES}}$)

Sumber: Philips Semiconductor. *PCD8544 48 x 84 Pixels Matrix LCD Controller/Driver Datasheet*. 15 Februari 2006. p.13
<<http://serdisplib.sourceforge.net/ser/doc/pcd8544.pdf>>

2.5.8. Fungsi *Reset*

Setelah *reset*, PCD 8544 memiliki kondisi sebagai berikut:

- *Power-down mode* (*bit* PD = 1)
- Pengalamatan horisontal (*bit* V = 0), *normal instruction set* (*bit* H = 0)
- *Display blank* (*bit* E = D = 0)
- *Address counter* X₆ s/d X₀ = 0; Y₂ s/d Y₀ = 0
- *Temperature control mode* (TC₁ TC₀ = 0)
- *Bias system* (BS₂ s/d BS₀ = 0)
- V_{LCD} = 0, *HV generator* tidak aktif
- Setelah *power-on*, isi dari RAM dalam kondisi *undefined*.

2.5.9. *Function Set*

2.5.9.1. *Bit* PD

- Semua *output* LCD pada V_{SS} (*display off*).
- *Bias generator* dan V_{LCD} *generator off*, V_{LCD} dapat tidak dihubungkan (*disconnected*).
- Osilator *off* (memungkinkan untuk eksternal *clock*).
- *Bus serial*, *command*, dan fungsi lain.
- Sebelum memasuki mode *Power-down*, RAM harus diisi dengan '0' untuk menjamin konsumsi arus yang spesifik.

2.5.9.2. *Bit* V

- Ketika V = 0, berarti dipilih mode pengalamatan secara horisontal dan data ditulis ke dalam DDRAM atau LCD.
- Ketika V = 1, berarti dipilih pengalamatan secara vertikal dan data ditulis ke dalam DDRAM atau LCD.

2.5.9.3. *Bit* H

- Ketika H = 0, perintah '*display control*', '*set Y address*' dan '*set X address*' dapat ditampilkan
- Ketika H = 1, perintah lain dapat dieksekusi.

Baik perintah '*write data*' dan '*function set*' dapat dieksekusi pada kedua kondisi tersebut.

2.5.10. *Display Control*

2.5.10.1. *Bit D dan E*

Bit D dan E digunakan untuk memilih *mode* dari tampilan.

2.5.11. *Set Alamat Y dari RAM*

Y_n merupakan pengalamatan vektor Y dari DDRAM. Berikut ini adalah tabel dari pengalamatan vektor Y.

Tabel 2.8. Pengalamatan Vektor Y

Y_2	Y_1	Y_0	Bank
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5

Sumber: Philips Semiconductor. *PCD8544 48 x 84 Pixels Matrix LCD Controller/Driver Datasheet*. 15 Februari 2006. p.15
<<http://serdisplib.sourceforge.net/ser/doc/pcd8544.pdf>>

2.5.12. *Set Alamat X dari RAM*

Alamat X merupakan alamat dari kolom. *Range* dari alamat X ini adalah 0 s/d 83 (53_H).

2.5.13. *Temperature Control*

Koefisien suhu dari V_{LCD} dapat diatur melalui *bit-bit* TC_1 dan TC_0 .

2.5.14. *Bias Value*

Level tegangan dari bias di-*set* dalam rasio dari R-R-nR-R-R, memberikan $1/(n + 4)$ sistem bias. *Multiplex rate (mux rate)* yang berbeda memerlukan faktor n

yang berbeda pula. Berikut ini adalah tabel yang menunjukkan cara dari pemrograman *bias system* dan tabel dari tegangan bias LCD.

Tabel 2.9. *Programming Required Bias System*

BS ₂	BS ₁	BS ₀	n	Recommended Mux Rate
0	0	0	7	1:100
0	0	1	6	1:80
0	1	0	5	1:65/1:65
0	1	1	4	1:48
1	0	0	3	1:40/1:34
1	0	1	2	1:24
1	1	0	1	1:18/1:16
1	1	1	0	1:10/1:9/1:8

Sumber: Philips Semiconductor. *PCD8544 48 x 84 Pixels Matrix LCD Controller/Driver Datasheet*. 15 Februari 2006. p.16
<<http://serdisplib.sourceforge.net/ser/doc/pcd8544.pdf>>

Tabel 2.10. *LCD Bias Voltage*

Symbol	Bias Voltage	Bias Voltage For 1/8 Bias
V1	V _{LCD}	V _{LCD}
V2	$(n+3)/(n+4)$	$7/8 \times V_{LCD}$
V3	$(n+2)/(n+4)$	$6/8 \times V_{LCD}$
V4	$2/(n+4)$	$2/8 \times V_{LCD}$
V5	$1/(n+4)$	$1/8 \times V_{LCD}$
V6	V _{SS}	V _{SS}

Sumber: Philips Semiconductor. *PCD8544 48 x 84 Pixels Matrix LCD Controller/Driver Datasheet*. 15 Februari 2006. p.16
<<http://serdisplib.sourceforge.net/ser/doc/pcd8544.pdf>>

Untuk Mux 1 : 48, nilai optimum dari nilai bias n, menghasilkan 1/8 bias, diperoleh dari: $n = \sqrt{48} - 3 = 3.928 = 4$.

2.5.15. Set Nilai V_{OP}

Tegangan operasi (V_{LCD}) dapat diatur melalui *software*. Nilai dari V_{LCD} tergantung dari LCD yang digunakan. Untuk mengatur V_{LCD} dengan *software* diberikan melalui rumus:

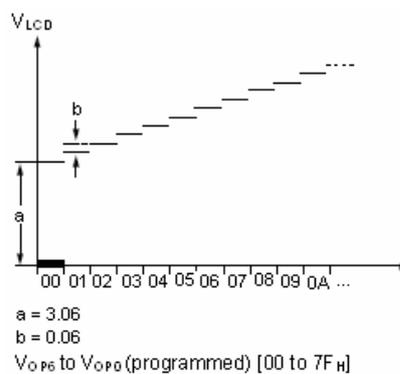
$$V_{LCD} = a + (V_{OP6} \text{ s/d } V_{OP0}) \times b \text{ [V]} \dots \dots \dots (2.1)$$

Pada PCD 8544, $a = 3.06$ dan $b = 0.06$ sehingga dari nilai a dan b tersebut memberikan memberikan *range* pada program mulai dari 3.00 s/d 10.68 pada suhu ruang.

Sebagai contoh, untuk Mux 1 : 48, nilai optimum dari tegangan operasi LCD dapat dihitung seperti berikut:

$$V_{LCD} = \frac{1 + \sqrt{48}}{\sqrt{2 \cdot \left(1 - \frac{1}{\sqrt{48}}\right)}} \cdot V_{th} = 6.06 \cdot V_{th} \dots\dots\dots(2.2)$$

Di mana V_{th} adalah tegangan *threshold* dari material LCD yang digunakan. Grafik berikut ini menjelaskan tentang pemrograman V_{OP} .



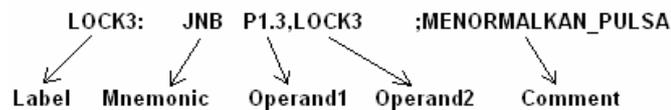
Gambar 2.27. Grafik Pemrograman V_{OP}

Sumber: Philips Semiconductor. *PCD8544 48 x 84 Pixels Matrix LCD Controller/Driver Datasheet*. 15 Februari 2006. p.16
 <<http://serdisplib.sourceforge.net/ser/doc/pcd8544.pdf>>

2.6. Program Sumber Assembly

Program sumber *assembly* merupakan program yang ditulis oleh pembuat program yang berupa kumpulan baris-baris perintah dan biasanya disimpan dengan ekstensi “.ASM”. Program sumber *assembly* ini ditulis menggunakan perangkat lunak teks editor seperti *notepad*, *editor* DOS, dan lain sebagainya.

Program *assembly* ini terdiri dari beberapa bagian yaitu label, *mnemonic*, *operand1*, *operand2*, dan *comment*. Gambar berikut ini menjelaskan bentuk dari program *assembly*.



Gambar 2.28. Bentuk Program *Assembly*

2.6.1. Label

Dalam suatu program, seringkali terjadi lompatan-lompatan atau percabangan ke suatu alamat tertentu maupun ke suatu baris tertentu. Selain itu, sering juga terdapat penambahan program atau pembuat program ingin menyisipkan suatu instruksi pada baris tertentu. Hal ini dapat dimungkinkan dengan adanya label. Label ini akan memudahkan pembuat program untuk melakukan hal-hal tersebut.

Di samping untuk memudahkan dalam membuat program, label ini juga dapat memudahkan seorang pemrogram untuk melacak jalannya program yang dibuat dengan lebih mudah.

Untuk membuat label, ada beberapa persyaratan yang harus dipenuhi dimana persyaratan ini kadang-kadang juga tergantung pada program *assembler* yang digunakan, yaitu:

1. Diawali dengan huruf.
2. Tidak diperbolehkan adanya label yang sama dalam suatu program *assembly*.
3. Maksimal 16 karakter.
4. Tidak diperbolehkan adanya karakter spasi dalam label.

2.6.2. *Mnemonic*

Mnemonic atau bisa juga disebut kode operasi adalah kode-kode yang akan dikerjakan oleh program *assembler* yang ada pada mikrokontroler. Ada berbagai macam *mnemonic*, antara lain yaitu *mov*, *add*, *cjne*, dan lain sebagainya.

Mnemonic-mnemonic ini memiliki waktu eksekusi dan besar jumlah *byte* yang bermacam-macam. Seperti contohnya *mnemonic* JB memiliki waktu eksekusi dua siklus dan jumlah *byte* tiga, CLR memiliki waktu eksekusi satu siklus dan jumlah *byte* satu, dan lain sebagainya.

2.6.3. *Operand*

Operand merupakan pelengkap dari *mnemonic*. Jumlah *operand* yang dibutuhkan oleh sebuah *mnemonic* tidak selalu sama, sebuah *mnemonic* dapat memiliki satu, dua, atau tiga dan bahkan tidak memiliki *operand* seperti yang terlihat pada contoh berikut:

CJNE R5,#10,ULANG	;memiliki tiga operand.
MOV R6,#0	;memiliki dua operand.
CLR C	;memiliki satu operand.
NOP	;tidak memiliki operand.

2.6.4. *Comment*

Bagian *comment* ini tidak mutlak ada dalam suatu program, namun bagian ini seringkali dibutuhkan untuk menjelaskan proses-proses kerja ataupun catatan-catatan tertentu pada bagian-bagian program. Bahkan pembuat program sekalipun seringkali membutuhkannya untuk mengingat kembali jalannya program yang dibuatnya.

Pembuatan *comment*, biasanya diawali dengan tanda “;” dan dapat diletakkan pada bagian manapun dari suatu program. Sebuah *comment* tidak hanya digunakan untuk menjelaskan satu baris perintah saja, akan tetapi juga dapat digunakan untuk menjelaskan kinerja dari beberapa perintah atau memberikan catatan-catatan tertentu yang dibutuhkan.

2.7. Program Assembler

Program *assembler* merupakan perangkat lunak yang dibutuhkan untuk melakukan proses *assembly* yang mengubah program sumber *assembly* menjadi program obyek atau *assembly listing*.

2.7.1. Program Obyek

Program obyek adalah hasil utama dari sebuah proses *assembly* berupa kode-kode yang hanya dikenali oleh mikroprosesor/mikrokontroler. Semua program obyek berupa kode heksa (bisa juga dilihat dalam angka biner) dan format penyimpanan *file* program (*executable*) ini disimpan dalam file dengan

ekstensi “.HEX” atau “.BIN”. *File* inilah yang nantinya dapat diisikan ke memori dari sebuah mikrokontroler setelah proses *assembly* dilakukan.

2.7.2. *Assembly Listing*

Assembly listing merupakan hasil dari proses *assembly* dalam rupa campuran dari program obyek, program sumber *assembly*, dan alamat-alamatnya. *Assembly listing* disimpan dalam *file* dengan ekstensi “.LST”.

2.7.3. Program *Downloader*

Program ini merupakan perangkat lunak yang dibutuhkan untuk *download* program obyek ke dalam memori. Program ini biasanya digunakan pada sistem mikrokontroler berupa *development system*.