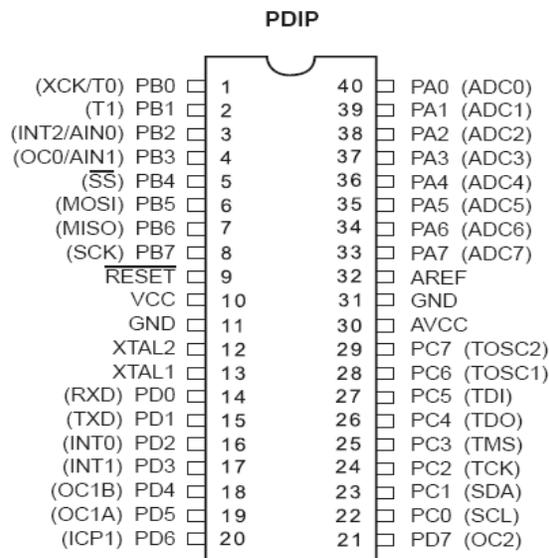


BAB 2

LANDASAN TEORI

2.1 Mikrokontroler AVR ATmega 32L

Mikrokontroler merupakan suatu *device* yang didalamnya sudah terintegrasi dengan I/O Port, RAM, ROM, sehingga dapat digunakan untuk berbagai keperluan kontrol. Mikrokontroler AVR ATmega 32L merupakan *low power* CMOS Mikrokontroler 8-bit yang dikembangkan oleh Atmel dengan arsitektur RISC (*Reduced Instruction Set Computer*) sehingga dapat mencapai *throughput* eksekusi instruksi 1 MIPS (*Million Instruction Per Second*). Mikrokontroler AVR dapat dikelompokkan menjadi 4 kelas yaitu kelas ATtiny, kelas AT90xx, keluarga ATmega, dan kelas AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, *peripheral*, *speed*, operasi tegangan, dan fungsinya sedangkan dari segi arsitektur dan instruksi yang digunakan bisa dikatakan hampir sama.



Gambar 2.1 Konfigurasi Pin Mikrokontroler AVR ATmega 32L

Penjelasan konfigurasi pin pada mikrokontroler AVR ATmega 32L secara umum:

- a. Pin 1 sampai 8 (*Port B*) merupakan *port parallel* 8 bit dua arah (*bidirectional*), yang dapat digunakan untuk *general purpose* dan *special feature*
- b. Pin 9 (*Reset*) jika terdapat minimum *pulse* pada saat *active low*
- c. Pin 10 (VCC) dihubungkan ke Vcc (2,7 – 5,5 Volt)
- d. Pin 11 dan 31 (GND) dihubungkan ke Vss atau *Ground*
- e. Pin 12 (XTAL 2) adalah pin masukkan ke rangkaian osilator *internal*. Sebuah osilator kristal atau sumber osilator luar dapat digunakan.
- f. Pin 13 (XTAL 1) adalah pin keluaran ke rangkaian osilator *internal*. Pin ini dipakai bila menggunakan osilator kristal.
- g. Pin 14 sampai 21 (*Port D*) adalah 8-bit dua arah (*bi-directional I/O port* dengan *internal pull-up resistors*) digunakan untuk *general purpose* dan *special feature*.
- h. Pin 22 sampai 29 (*Port C*) adalah 8-bit dua arah (*bi-directional I/O port* dengan *internal pull-up resistors*) digunakan untuk *general purpose* dan *special feature*.
- i. Pin 30 adalah Avcc pin penyuplai daya untuk *port A* dan *A/D converter* dan dihubungkan ke Vcc. Jika ADC digunakan maka pin ini dihubungkan ke Vcc.
- j. Pin 32 adalah A REF pin yang berfungsi sebagai referensi untuk pin analog jika *A/D Converter* digunakan.

- k. Pin 33 sampai 40 (*Port A*) adalah 8-bit dua arah (*bi-directional I/O*) port dengan *internal pull-up resistors* digunakan untuk *general purpose*.

Penjelasan konfigurasi pin pada mikrokontroler AVR ATmega 32L yang mempunyai fungsi khusus yaitu:

- a. Pin 33 sampai 40 (*Port A*) dapat digunakan sebagai,

Tabel 2.1 Tabel Fungsi Khusus *Port A*

PA0	<i>Input ADC PA0</i>
PA1	<i>Input ADC PA1</i>
PA2	<i>Input ADC PA2</i>
PA3	<i>Input ADC PA3</i>
PA4	<i>Input ADC PA4</i>
PA5	<i>Input ADC PA5</i>
PA6	<i>Input ADC PA6</i>
PA7	<i>Input ADC PA7</i>

- b. Pin 1 sampai 8 (*Port B*) dapat digunakan sebagai,

Tabel 2.2 Tabel Fungsi Khusus *Port B*

PB7	SCK (<i>SPI Bus Serial Clock</i>)
PB6	MISO (<i>SPI Bus Master Input/Slave Output</i>)
PB5	MOSI (<i>SPI Bus Master Output/Slave Input</i>)
PB4	SS (<i>SPI Slave Select Input</i>)
PB3	AIN1 (<i>Analog Comparator Negative Input</i>), OC0 (<i>Timer/Counter0 Output Compare</i>)

	<i>Match Output</i>)
PB2	AIN0 (<i>Analog Comparator Positive Input</i>), INT2 (<i>External Interrupt 2 Input</i>)
PB1	T1 (<i>Timer/Counter1 External Counter Input</i>)
PB0	T0 (<i>Timer/Counter0 External Counter Input</i>), XCK (<i>USART External Clock Input/Output</i>)

- c. Pin 22 sampai 29 (*Port C*) dapat digunakan sebagai,

Tabel 2.3 Tabel Fungsi Khusus *Port C*

PC7	TOSC2 (<i>Timer Oscillator Pin 2</i>)
PC6	TOSC1 (<i>Timer Oscillator Pin 1</i>)
PC5	TDI (<i>JTAG Test Data In</i>)
PC4	TDO (<i>JTAG Test Data Out</i>)
PC3	TMS (<i>JTAG Test Mode Select</i>)
PC2	TCK (<i>JTAG Test Clock</i>)
PC1	SDA (<i>Two-wire Serial Bus Data Input/Output Line</i>)
PC0	SCL (<i>Two-wire Serial Bus Clock Line</i>)

- d. Pin 14 sampai 21 (*Port D*) dapat digunakan sebagai,

Tabel 2.4 Tabel Fungsi Khusus *Port D*

PD7	OC2 (<i>Timer/Counter2 Output Compare Match Output</i>)
-----	---

PD6	ICP1 (<i>Timer/Counter1 Input Capture Pin</i>)
PD5	OC1A (<i>Timer/Counter1 Output Compare A Match Output</i>)
PD4	OC1B (<i>Timer/Counter1 Output Compare B Match Output</i>)
PD3	INT1 (<i>External Interrupt 1 Input</i>)
PD2	INT0 (<i>External Interrupt 0 Input</i>)
PD1	TXD (<i>USART Output Pin</i>)
PD0	RXD (<i>USART Input Pin</i>)

Keistimewaan yang terdapat pada Mikrokontroler AVR ATmega 32L :

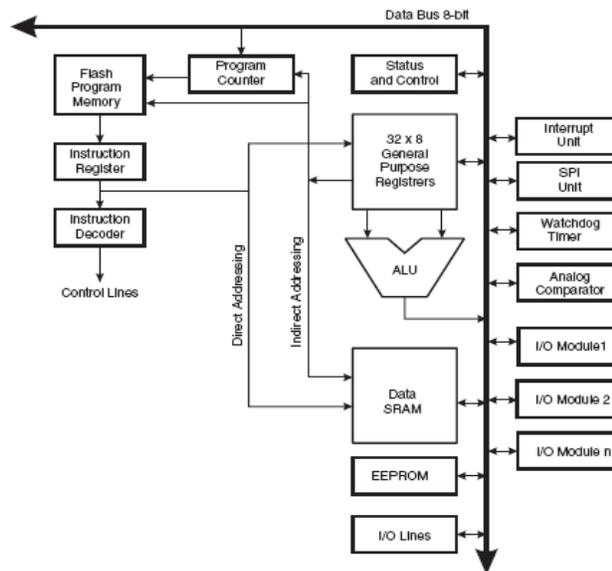
- 8-bit CPU sebagai pusat pengendalian aplikasi
- Mempunyai 131 instruksi yang sebagian besar dieksekusi dalam 1 *cycle*
- 32 *register* umum yang terhubung dengan ALU (*Arithmetic Logic Unit*)
- Kemampuan memproses instruksi sampai 8 MIPS (*million instruction per second*)
- Memiliki 32 Kbyte untuk *Flash* dalam untuk menyimpan program dan dapat ditulis ulang hingga 10.000 kali
- Memiliki 1024 Bytes EEPROM dengan *endurance* : 100,000 *Write/Erase Cycles*
- Memiliki 2 KByte Internal SRAM (*Static Random Access Memory*) digunakan untuk menyimpan sementara data dari program *flash*

- ADC (*Analog To Digital Converter*) internal dengan resolusi 10 bit sebanyak 8 *channel*
- 32 jalur I/O (*Input/Output*) yang terpisah dalam empat *port* yaitu A, *port* B, *port* C, dan *port* D
- 16 bit *timer/counter* dan 8 bit *timer/counter*
- *Full Duplex Universal Synchronous Asynchronous Receiver Transmitter (USART)*
- RTC (*Real Time Clock*) dengan osilator terpisah
- SPI (*Serial Peripheral Interface*) untuk komunikasi serial yang memiliki kecepatan yang relatif tinggi pada jarak dekat
- Enam pilihan mode *sleep* menghemat penggunaan daya listrik
- *Watchdog timer* yang dapat diprogram dengan osilator internal
- Dapat beroperasi pada tegangan 2,7 – 5,5 V
- Konsumsi daya :
 - 1.1mA ketika aktif
 - 0.35mA ketika *Idle*
 - *Power-down* < 1 μ A

2.1.1 Arsitektur ATmega 32L

Untuk meningkatkan kemampuan, mikrokontroler AVR ATmega 32L menggunakan teknologi RISC (*Reduced Instruction Set Computer*) di mana set instruksi dikurangi lebarnya sehingga semua instruksi mempunyai panjang 16 bit dan sebagian besar instruksi dieksekusi dalam *single clock*, serta pengurangan kompleksitas pengalaman. Mikrokontroler AVR menggunakan arsitektur

Harvard dengan memisahkan memori dan jalur *bus* untuk program dan data agar meningkatkan kemampuan karena dapat mengakses program memori dan data memori secara bersamaan. Mikrokontroler AVR memiliki *fast access register file* dengan 32 *register* x 8 bit. Dengan 32 *register* AVR dapat mengeksekusi beberapa instruksi sekali jalan (*single cycle*). 6 dari 32 *register* yang ada dapat digunakan sebagai *indirect address register pointer* 16 bit untuk pengalamatan *data space*, yang memungkinkan penghitungan alamat yang efisien.



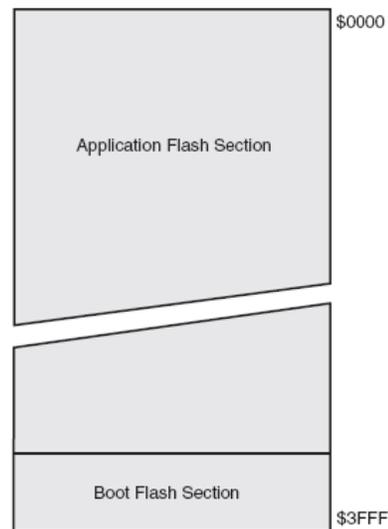
Gambar 2.2 Arsitektur Mikrokontroler AVR ATmega 32L
(Datasheet ATMEL ATmega 32L, 2008)

2.1.2 Memori ATmega 32L

Pada mikrokontroler AVR terdapat dua ruang memori utama yaitu memori data SRAM (*Static Random Access Memory*) dan memori program ISP (*In – System Reprogrammable Flash Program Memory*). Selain dua memori utama, pada ATmega 32L terdapat memori EEPROM untuk penyimpanan data sebesar 1KByte.

A. *Flash Memory*

ATmega 32L memiliki 32 Kbyte *On – Chip In – System Reprogrammable Flash Memory* yang digunakan untuk menyimpan program dan menyimpan vektor interupsi. Karena semua instruksi pada AVR mempunyai lebar instruksi 16 atau 32 bit, maka ATmega 32L memiliki organisasi memori 16 KByte x 16 bit dengan alamat dari \$0000 sampai \$3FFF. Untuk keamanan *software*, ruang *flash program memory* dibagi menjadi dua bagian, *Boot Program* dan *Application Program*, ATmega 32L memiliki *Program Counter* (PC) dengan lebar 14 bit untuk mengamati *program memory* sebesar 16 KByte.

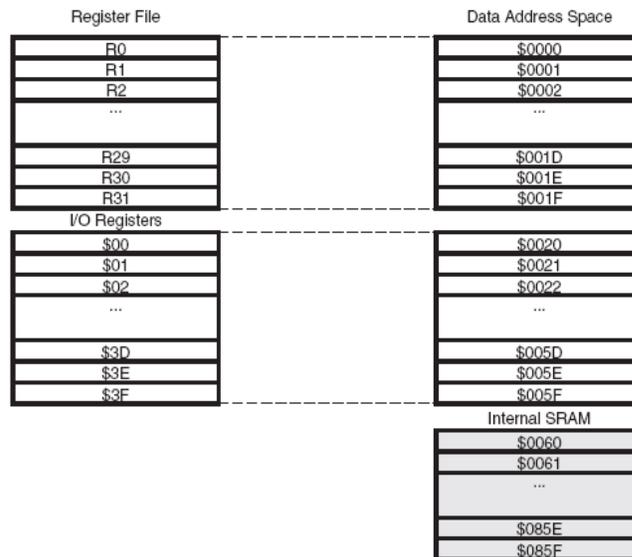


Gambar 2.3 *Program Memory* ATmega 32L

B. *SRAM (Static Random Access Memory)*

SRAM (*Static Random Access Memory*) atau biasa disebut juga *data memory* yang berfungsi untuk menyimpan data sementara, SRAM merupakan memori yang termasuk golongan *volatile* yang berarti data akan

hilang ketika tidak mendapat sumber listrik dan SRAM tidak membutuhkan *refresh* secara periodik dikarenakan SRAM menggunakan teknologi transistor. Pada ATmega 32L terdapat tiga bagian pada data memori yaitu, *Register file* untuk *register* R0 sampai R31 dengan alamat data dari \$0000 sampai dengan \$001F, *I/O Register* dengan alamat data dari \$0020 sampai dengan \$005F, dan Internal SRAM dengan alamat dari \$0060 sampai dengan \$085F. Pada ATmega 32L memiliki Internal SRAM sebesar 2 KByte.



Gambar 2.4 SRAM Peta *Memory* ATmega 32L

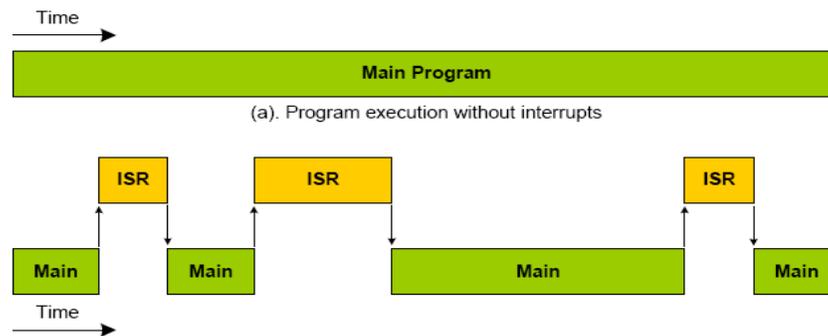
C. EEPROM (*Electrically Erasable Programmable Read-Only Memory*)

EEPROM secara umum digunakan untuk menyimpan data yang tetap, EEPROM termasuk golongan memori *non – volatile* yang berarti data tidak akan hilang walaupun EEPROM tersebut kehilangan sumber listrik. Pada

ATmega 32L terdapat EEPROM sebesar 1KByte yang dapat diakses *read* / *write* sesuai kebutuhan.

2.1.3 Interupsi

Interupsi adalah kondisi yang mengharuskan mikrokontroler menghentikan sementara eksekusi program utama dan mengeksekusi rutin *interrupt* / *Interrupt Service Routine (ISR)*, setelah melaksanakan ISR secara lengkap, maka mikrokontroler akan kembali melanjutkan eksekusi program utama yang dihentikan.



Gambar 2.5 Interupsi

Untuk menggunakan interupsi pada ATmega 32L terdapat tiga *register* yaitu, MCUR, GICR, dan SREG yang masing-masing berfungsi untuk mengatur kondisi dan cara pengaktifan interupsi dan memiliki lebar 8 bit.

- *Register* MCUR / MCU Control Register merupakan *register* yang menentukan tipe interupsi eksternal dengan lebar 8 bit.

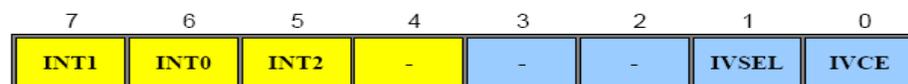


MCUCR / MCU Control Register

Bit	Symbol	Fuction																																				
7,5,4	SM2..0	Sleep Mode Select Bits. Digunakan untuk memilih mode sleep MCU																																				
		<table border="1"> <tr> <td>SM2</td> <td>SM1</td> <td>SM0</td> <td>Sleep Mode</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Idle</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>ADC Noise Reduction</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Power-down</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Power-save</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Reserved</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Reserved</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Standby</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Extended Standby</td> </tr> </table>	SM2	SM1	SM0	Sleep Mode	0	0	0	Idle	0	0	1	ADC Noise Reduction	0	1	0	Power-down	0	1	1	Power-save	1	0	0	Reserved	1	0	1	Reserved	1	1	0	Standby	1	1	1	Extended Standby
SM2	SM1	SM0	Sleep Mode																																			
0	0	0	Idle																																			
0	0	1	ADC Noise Reduction																																			
0	1	0	Power-down																																			
0	1	1	Power-save																																			
1	0	0	Reserved																																			
1	0	1	Reserved																																			
1	1	0	Standby																																			
1	1	1	Extended Standby																																			
6	SE	Sleep Enable. Digunakan untuk mengaktifkan mode Sleep dengan memberikan logika 1 pada SE sebelum Sleep dijalankan. Setelah Wake-up harus segera diberi logika 0.																																				
3,2	ISC11, ISC10	Interrupt Sense Control 1 Bit. Digunakan untuk memilih jenis pulsa trigger External Interrupt 1 (INT1).																																				
		<table border="1"> <tr> <td>ISC11</td> <td>ISC10</td> <td>Description</td> </tr> <tr> <td>0</td> <td>0</td> <td>The low level of INT1 generates an interrupt request.</td> </tr> <tr> <td>0</td> <td>1</td> <td>Any logical change on INT1 generate an interrupt request</td> </tr> <tr> <td>1</td> <td>0</td> <td>The falling edge of INT1 generate an interrupt request.</td> </tr> <tr> <td>1</td> <td>1</td> <td>The rising edge of INT1 generate an interrupt request.</td> </tr> </table>	ISC11	ISC10	Description	0	0	The low level of INT1 generates an interrupt request.	0	1	Any logical change on INT1 generate an interrupt request	1	0	The falling edge of INT1 generate an interrupt request.	1	1	The rising edge of INT1 generate an interrupt request.																					
ISC11	ISC10	Description																																				
0	0	The low level of INT1 generates an interrupt request.																																				
0	1	Any logical change on INT1 generate an interrupt request																																				
1	0	The falling edge of INT1 generate an interrupt request.																																				
1	1	The rising edge of INT1 generate an interrupt request.																																				
1,0	ISC01, ISC00	Interrupt Sense Control 0 Bit. Digunakan untuk memilih jenis pulsa trigger External Interrupt 0 (INT0).																																				
		<table border="1"> <tr> <td>ISC01</td> <td>ISC00</td> <td>Description</td> </tr> <tr> <td>0</td> <td>0</td> <td>The low level of INT0 generates an interrupt request.</td> </tr> <tr> <td>0</td> <td>1</td> <td>Any logical change on INT0 generate an interrupt request</td> </tr> <tr> <td>1</td> <td>0</td> <td>The falling edge of INT0 generate an interrupt request.</td> </tr> <tr> <td>1</td> <td>1</td> <td>The rising edge of INT0 generate an interrupt request.</td> </tr> </table>	ISC01	ISC00	Description	0	0	The low level of INT0 generates an interrupt request.	0	1	Any logical change on INT0 generate an interrupt request	1	0	The falling edge of INT0 generate an interrupt request.	1	1	The rising edge of INT0 generate an interrupt request.																					
ISC01	ISC00	Description																																				
0	0	The low level of INT0 generates an interrupt request.																																				
0	1	Any logical change on INT0 generate an interrupt request																																				
1	0	The falling edge of INT0 generate an interrupt request.																																				
1	1	The rising edge of INT0 generate an interrupt request.																																				

Gambar 2.6 MCUR / MCU Control Register

- Register GICR / General Interrupt Control Register berfungsi untuk mengatur tipe interupsi internal register GICR / General Interrupt Control Register mempunyai lebar 8 bit.



GICR / General Interrupt Control Register

Bit	Symbol	Fuction
7	INT1	External Interrupt Request 1 Enable. Digunakan untuk mengaktifkan INT 1 dengan syarat bit I pada SREG juga diset .
6	INT0	External Interrupt Request 0 Enable. Digunakan untuk mengaktifkan INT0 dengan syarat bit I pada SREG juga diset .
5	INT1	External Interrupt Request 2 Enable. Digunakan untuk mengaktifkan INT2 dengan syarat bit I pada SREG juga diset .
4,3,2	-	Reserved
1	IVSEL	Interrupt Vector Select. Jika IVSEL = 0 maka interrupt vector ditempatkan pada start address flash memory Jika IVSEL = 1 maka interrupt vector ditempatkan pada awal bootloader dalam flash memory.
1	IVCE	Interrupt Vector Change Enable . Digunakan untuk mengaktifkan perubahan interrupt vector .

Gambar 2.7 GICR / General Interrupt Control Register

Pada mikrokontroler ATmega 32L terdapat 21 interupsi vektor seperti,

Vector No.	Program Address ⁽²⁾	Source	Interrupt Definition
1	\$000 ⁽¹⁾	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	\$002	INT0	External Interrupt Request 0
3	\$004	INT1	External Interrupt Request 1
4	\$006	INT2	External Interrupt Request 2
5	\$008	TIMER2 COMP	Timer/Counter2 Compare Match
6	\$00A	TIMER2 OVF	Timer/Counter2 Overflow
7	\$00C	TIMER1 CAPT	Timer/Counter1 Capture Event
8	\$00E	TIMER1 COMPA	Timer/Counter1 Compare Match A
9	\$010	TIMER1 COMPB	Timer/Counter1 Compare Match B
10	\$012	TIMER1 OVF	Timer/Counter1 Overflow
11	\$014	TIMER0 COMP	Timer/Counter0 Compare Match
12	\$016	TIMER0 OVF	Timer/Counter0 Overflow
13	\$018	SPI, STC	Serial Transfer Complete
14	\$01A	USART, RXC	USART, Rx Complete
15	\$01C	USART, UDRE	USART Data Register Empty
16	\$01E	USART, TXC	USART, Tx Complete
17	\$020	ADC	ADC Conversion Complete
18	\$022	EE_RDY	EEPROM Ready
19	\$024	ANA_COMP	Analog Comparator
20	\$026	TWI	Two-wire Serial Interface
21	\$028	SPM_RDY	Store Program Memory Ready

Gambar 2.8 Tabel Interupsi Vektor Pada ATmega 32L

2.1.4 USART (*Universal Synchronous and Asynchronous serial Receiver and Transmitter*)

Serial USART (*Universal Synchronous and Asynchronous serial Receiver and Transmitter*) merupakan salah fitur yang telah disediakan mikrokontroler AVR untuk berkomunikasi serial dengan mode asinkron. Untuk melakukan komunikasi serial UART, maka terdapat *register-register* yang harus diset nilainya seperti *register* UBRR (USART *Baud Rate Register*), UCSRB (USART *Control and Status Register B*), dan UCSRC (USART *Control and Status Register C*).

- UBRR merupakan *register* 16 bit yang berfungsi untuk menentukan kecepatan transmisi data. UBRR dibagi menjadi dua, yaitu UBRRH dan UBBRL.

Bit	15	14	13	12	11	10	9	8	
	URSEL		-	-	-	UBRR[11:8]			UBRRH
	UBRR[7:0]								UBRRL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Gambar 2.9 Register UBRR

URSEL adalah bit untuk pemilihan akses UBRR dan UCSRC. Set 0 untuk akses UBRR, hal ini dikarenakan UBRRH dan UCSRC menggunakan lokasi I/O yang sama.

- UBRR_{0..11} adalah untuk menyimpan konstanta kecepatan komunikasi serial (*baud rate*), Untuk mengisi nilai *baud rate* digunakan rumus.

Operating Mode	Equation for Calculating Baud Rate ⁽¹⁾	Equation for Calculating UBRR Value
Asynchronous Normal Mode (U2X = 0)	$BAUD = \frac{f_{OSC}}{16(UBRR + 1)}$	$UBRR = \frac{f_{OSC}}{16BAUD} - 1$
Asynchronous Double Speed Mode (U2X = 1)	$BAUD = \frac{f_{OSC}}{8(UBRR + 1)}$	$UBRR = \frac{f_{OSC}}{8BAUD} - 1$
Synchronous Master Mode	$BAUD = \frac{f_{OSC}}{2(UBRR + 1)}$	$UBRR = \frac{f_{OSC}}{2BAUD} - 1$

Note: 1. The baud rate is defined to be the transfer rate in bit per second (bps).

BAUD Baud rate (in bits per second, bps)

f_{OSC} System Oscillator clock frequency

Gambar 2.10 Perhitungan Nilai Baud Rate

- Register USCRB adalah register yang digunakan untuk mengaktifkan penerimaan dan pengiriman data USART.

Bit	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCS22	RXB8	TXB8	USCRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.11 Register USCRB

RXEN : Jika di set 1 maka akan mengaktifkan penerimaan

TXEN : Jika di set 1 maka akan mengaktifkan pengiriman

RXCIE : Jika di set 1 maka akan mengaktifkan interupsi penerimaan

TXCIE : Jika di set 1 maka akan mengaktifkan interupsi pengiriman

- *Register USCRC* adalah *register* yang digunakan untuk mengatur mode komunikasi USART.

Bit	7	6	5	4	3	2	1	0	
	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	USCRC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	0	0	0	0	1	1	0	

Gambar 2.12 Register USCRC

URSEL : Jika di set 1 maka *register* USCRC bisa diakses, sebab alamat *register* USCRC dan UBRR sama

UCSZ2...UCSZ0 : Menentukan ukuran karakter yang dikirimkan

Jika UCSZ2...UCSZ0 = 000 maka ukuran karakter 5 bit

Jika UCSZ2...UCSZ0 = 001 maka ukuran karakter 6 bit

Jika UCSZ2...UCSZ0 = 010 maka ukuran karakter 7 bit

Jika UCSZ2...UCSZ0 = 011 maka ukuran karakter 8 bit

Jika UCSZ2...UCSZ0 = 100-110 tidak digunakan

Jika UCSZ2...UCSZ0 = 111 maka ukuran karakter 9 bit

A. Pengiriman Data

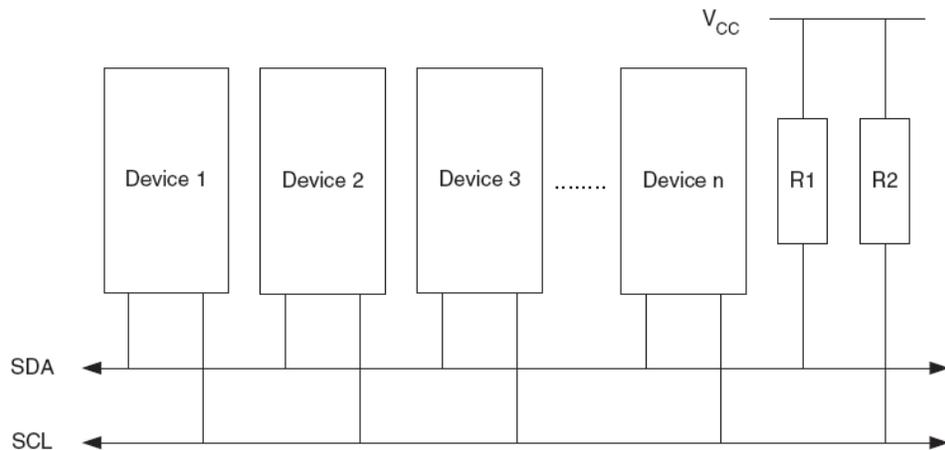
Pengiriman data dilakukan per byte menunggu UDR kosong (UDR = *register* tempat menyimpan data USART, menjadi satu dengan *register* UBRR). Jika kosong, maka bit UDRE (USART Data Register Empty) pada UCSRA akan set, sehingga siap menerima data baru yang akan dikirim.

B. Penerimaan Data

Penerimaan data dilakukan dengan memeriksa bit RXC (USART *Receive Complete*) pada *register* UCSRA (USART *Control and Status Register A*). RXC akan set 1 jika ada data yang siap dibaca. Data yang diterima akan disimpan pada *register* UDR.

2.2 Two-wire Serial Interface

Two-wire Serial Interface (TWI) atau secara umum dapat disebut I2C (*Inter-Integrated Circuit*), adalah protokol yang memperbolehkan *system designer* untuk menghubungkan hingga 128 *devices* berbeda menggunakan hanya TWI *bi-directional bus lines*, satu untuk *clock* (SCL) dan satu lagi untuk data (SDA). Satu-satunya *external hardware* yang dibutuhkan untuk mengimplementasi *bus*-nya adalah sebuah *pull-up resistor* untuk setiap jalur *bus* TWI. Semua *device* yang terhubung ke *bus* memiliki alamatnya sendiri, dan mekanisme untuk memecahkan permasalahan *bus* terdapat pada protokol TWI. Jenis komunikasi yang dilakukan antar peralatan dengan menggunakan protokol TWI mempunyai sifat *serial synchronous half duplex bi-directional*, dimana data yang ditransmisikan dan diterima hanya melalui satu jalur data SDA *line* (bersifat *serial*), setiap penggunaan jalur data bergantian antar perangkat (bersifat *half duplex*) dan data dapat ditransmisikan dari dan ke sebuah perangkat (bersifat *bi-directional*). (Datasheet ATMEL ATmega 32L, 2008)



Gambar 2.13 TWI Bus Interconnection

Untuk menggunakan *Two-wire Serial Interface* (TWI), terdapat beberapa istilah yaitu :

- *Master* yang mengatur untuk memulai dan menghentikan transmisi, dan *Master* selalu membangkitkan *SCL clock*
- *Slave* merupakan sebutan untuk *device* yang dialamat oleh *master*
- *Transmitter* merupakan *device* yang menempatkan data pada jalur *bus*
- *Receiver* merupakan *device* yang datanya dibaca dari *bus*

2.2.1 Electrical Interconnection

Kedua jalur SDA dan SCL merupakan *driver* yang bersifat "open drain", yang berarti bahwa IC yang digunakan dapat men-*drive output*-nya *low*, tetapi tidak dapat men-*drive* menjadi *high*. Untuk dapat mendapatkan data yang *high* maka kita harus menyediakan resistor *pull-up* pada tegangan *power supply* sebesar 5 volt terhadap jalur SDA dan SCL tersebut. Kita hanya membutuhkan *pull-up* resistor untuk semua jalur *I²C bus*, tidak untuk semua perangkat yang kita gunakan, pemasangan resistor *pull-up* dapat dilihat seperti pada gambar 2.13 TWI Bus Interconnection. Jika resistor-resistor tersebut tidak ada, maka jalur SCL dan

SDA akan selalu mendekati *low* – mendekati 0 volt dan jalur-jalur I²C *bus* tidak dapat bekerja. Nilai resistor yang secara umum digunakan berkisar dari 1K8Ω hingga 47K Ω. Biasanya nilai 1K8Ω, 4K7Ω dan 10KΩ merupakan nilai-nilai yang umum digunakan tetapi semua nilai yang berada dalam *range* nilai di atas dapat digunakan dan bekerja dengan baik.

Tabel 2.5 Tabel Nilai *Pull-up* Resistor Yang Digunakan Pada Jalur TWI

<i>Condition</i>	<i>Min</i>	<i>Max</i>
$f_{SCL} \leq 100 \text{ kHz}$	$\left(\frac{V_{CC} - 0.4V}{3mA}\right) \Omega$	$\left(\frac{1000ns}{C_b}\right) \Omega$
$f_{SCL} > 100 \text{ kHz}$	$\left(\frac{V_{CC} - 0.4V}{3mA}\right) \Omega$	$\left(\frac{300ns}{C_b}\right) \Omega$

Ket : C_b = kapasitas pada satu garis jalur dalam pF

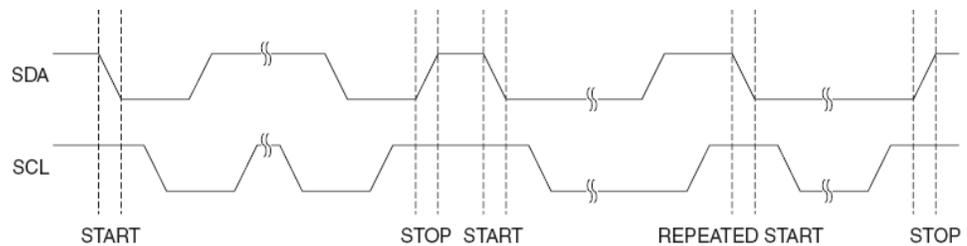
2.2.2 Mengirimkan data

Setiap bit data yang dikirim pada TWI *bus* didampingi dengan sebuah *pulse* pada jalur *clock*. Level tegangan pada jalur data harus stabil saat jalur *clock* dalam kondisi *high*. Satu-satunya pengecualian dari peraturan ini adalah untuk menghasilkan kondisi *Start* dan *Stop*.

A. Kondisi *Start* Dan *Stop*

Master memulai dan mengakhiri pengiriman data. Pengiriman dimulai saat *master* mengeluarkan kondisi *Start* pada *bus*, dan diakhiri pada saat *master* mengeluarkan kondisi *Stop*. Kondisi *Start* terjadi ketika SCL dalam kondisi *high* sementara SDA dalam kondisi *falling edge* (transisi *high* ke *low*). Kondisi *Stop* terjadi ketika SCL dalam kondisi *high* sementara SDA dalam kondisi *rising edge* (transisi *low* ke *high*). Di antara kondisi *Start* dan

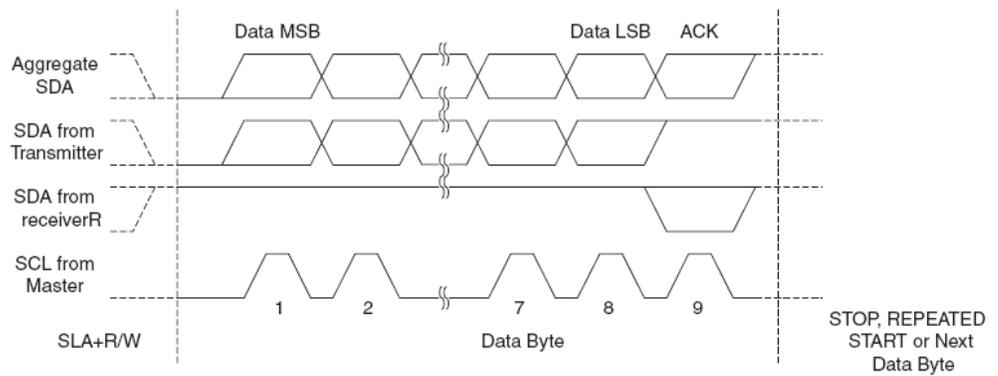
Stop, jalur dinyatakan sibuk dan tidak ada *master* lain yang mencoba untuk merampas *control bus*. Masalah khusus muncul saat sebuah kondisi *Start* baru muncul diantar kondisi *Start* dan *Stop*. Hal ini disebut kondisi *Repeated Start*, dan digunakan saat *master* menginginkan memulai pengiriman baru tanpa melepas *control bus*. Setelah sebuah *Repeated Start*, *bus* menjadi sibuk sampai *Stop* berikutnya.



Gambar 2.14 Kondisi *Start*, *Repeated Start*, Dan *Stop*
(Datasheet ATMEL ATmega 32L, 2008)

B. Format Paket Data

Semua paket data yang dikirim pada jalur TWI memiliki panjang Sembilan bit, berisikan satu byte data dan sebuah bit *acknowledge*. Selama sebuah transfer data, *master* menghasilkan *clock* dan kondisi *Start* dan *Stop*, saat penerima bertanggung jawab untuk men-*acknowledge* yang ditangkap. Sebuah *Acknowledge* (ACK) ditandai dengan penerima membuat jalur SDA *low* selama putaran SCL kesembilan. Jika penerima membiarkan jalur SDA *high* itu menandakan NACK. Saat penerima menerima byte terakhir, atau untuk sebab lain tidak dapat menerima byte lagi, penerima harus memberitahu pengirim dengan mengirimkan sebuah NACK setelah byte terakhir. MSB dari byte data dikirim pertama.



Gambar 2.15 Format Paket Data

2.2.3 Register TWI

Agar dapat menggunakan fitur TWI (*Two Wire Serial Interface*) yang sudah terdapat pada mikrokontroler ATmega 32L, maka terdapat beberapa *register* TWI yang harus diset nilainya.

A. TWI Bit Rate Register – TWBR

TWBR merupakan *register* yang memiliki lebar 8 bit dan berfungsi untuk membangkitkan *bitrate* dengan SCL *clock*, untuk menentukan frekuensi SCL maka dapat diperoleh dari persamaan SCL Frekuensi = CPU Clock Frequency / 16+2(TWBR). 4^{TWPS}

- TWBR = nilai dari *register* bit rate pada TWI
- TWPS = nilai dari bit prescaler bits pada TWSR

Bit	7	6	5	4	3	2	1	0	
	TWBR								
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.16 Register TWBR

B. TWI Control Register – TWCR

TWCR merupakan *register* yang berfungsi untuk mengatur operasi pada TWI, seperti TWI *Enable* untuk inialisasi kondisi *Start* dan *Stop*, untuk membangkitkan *Acknowledge*, untuk mendeteksi terjadinya *Collision*.

Bit	7	6	5	4	3	2	1	0	
	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	TWCR
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.17 Register TWCR

C. TWI Status Register (TWSR)

Register TWSR berfungsi untuk mewakili status dari TWI *logic* dan status *Two –wire serial bus*. Nilai yang dibaca dari *register* TWSR berisi nilai 5 bit status 1 bit *reserved* dan 2 bit nilai *prescaler*.

Bit	7	6	5	4	3	2	1	0	
	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	TWSR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	1	1	1	1	1	0	0	0	

Gambar 2.18 Register TWSR

D. TWI Data Register (TWDR)

Register TWDR dengan lebar 8 bit, pada saat mode pengiriman *register* ini berisi sebagai data selanjutnya, sedangkan pada mode penerimaan *register* ini sebagai berisi data terakhir.

Bit	7	6	5	4	3	2	1	0	
	TWD7	TWD6	TWD5	TWD4	TWD3	TWD2	TWD1	TWD0	TWDR
Read/Write	R/W								
Initial Value	1	1	1	1	1	1	1	1	

Gambar 2.19 Register TWDR

E. TWI (*Slave*) Address Register (TWAR)

Register TWAR berfungsi untuk menyimpan alamat *slave* dengan lebar 7 bit, dan bit LSB yang terdapat pada *register* TWAR berfungsi untuk melakukan *general call address*.

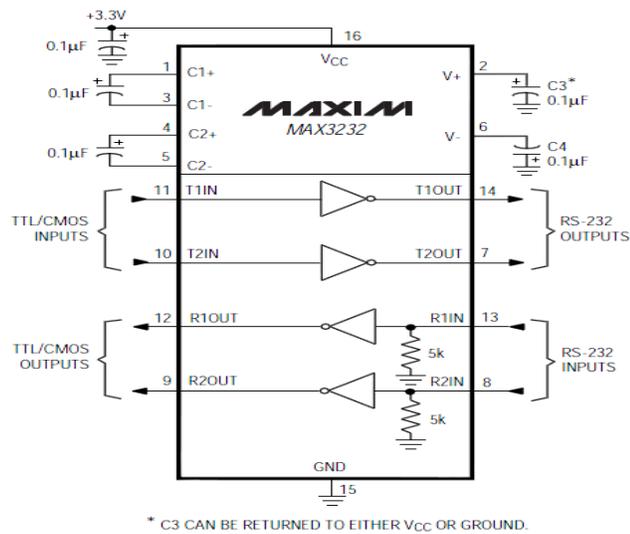
Bit	7	6	5	4	3	2	1	0	
	TWA6 TWA5 TWA4 TWA3 TWA2 TWA1 TWA0 TWGCE								TWAR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	1	1	1	1	1	1	1	0	

Gambar 2.20 Register TWAR

2.3 Standar Komunikasi RS-232

MAX-3232 adalah IC (*Integrated Circuit*) *dual driver* atau *receiver* yang meliputi sebuah pembangkit tegangan kapasitif untuk men-*supply* tingkat tegangan TIA/EIA-232-F dari sebuah *supply* tegangan 3.0V-5.5V. Setiap *receiver* merubah TIA/EIA-232-F yang masuk menjadi tingkat 5V TTL/CMOS. *Receiver* ini memiliki 1.2V *threshold*, 0.3V *hysteresis*, dan dapat menerima (-25)V s.d 25V *input*. Setiap *driver* merubah tingkat tegangan TTL/CMOS pada *input* menjadi tingkat tegangan TIA/EIA-232-F. IC MAX-3232 memiliki fungsi :

- Beroperasi dari sebuah *supply* tenaga 3.0V-5.5V dengan 0.1 μ F kapasitor *Charge-Pump*
- Bekerja hingga 120 Kbit/s
- Dua *driver* dan dua *receiver*
- -25V s.d 25V tingkat *input*
- Arus *supply* rendah sebesar 300 μ A



Gambar 2.21 Konfigurasi Standar RS-232

2.3.1 Komunikasi Serial Pada *Hardware*

Komunikasi serial adalah komunikasi yang tiap-tiap bit data dikirim secara berurutan dimulai dari LSB (*Least Significant Bit*) dan bertahap sampai MSB (*Most Significant Byte*) dalam satu waktu. Pada komunikasi serial terdapat dua mode:

A. Mode Serial Sinkron

Mode sinkron merupakan mode komunikasi yang pengiriman tiap bit data dilakukan dengan menggunakan sinkronisasi *clock*. Pada saat *transmitter* hendak mengirimkan data, harus disertai *clock* untuk sinkronisasi antara *transmitter* dengan *receiver*.

B. Mode Serial Asinkron

Mode asinkron merupakan mode komunikasi yang pengiriman tiap bit data dilakukan tanpa menggunakan sinkronisasi *clock*. *Transmitter* yang ingin mengirimkan data harus menyepakati suatu standar UART (*Universal Asynchronous Receive Transmit*) sehingga komunikasi data dilakukan

dengan suatu standar yang telah disepakati terlebih dahulu oleh *transmitter* dan *receiver*.

Pada komunikasi serial *port* dibagi menjadi 2 kelompok yaitu *Data Communication Equipment* (DCE) dan *Data Terminal Equipment* (DTE). Contoh dari DCE ialah *modem*, *plotter*, *scanner* dan lain lain sedangkan contoh dari DTE ialah *terminal* di komputer. Spesifikasi elektronik dari *serial port* merujuk pada *Electronic Industry Association* (EIA).

2.4 RFID (*Radio Frequency Identification Device*)

RFID adalah proses identifikasi objek dengan menggunakan frekuensi transmisi radio. Cara RFID mengidentifikasi objek nya adalah dengan menggunakan frekuensi radio untuk membaca informasi dari sebuah *device* kecil yang disebut *tag* atau *transponder*, dimana *transponder* itu sendiri terdiri dari *transmitter* dan *responder*. *Tag* RFID akan mengenali diri sendiri ketika mendeteksi sinyal dari *device* yang kompatibel, yaitu pembaca RFID (*micro-reader*).

RFID merupakan teknologi identifikasi yang fleksibel, mudah digunakan, dan sangat cocok untuk operasi otomatis. RFID mengkombinasikan keunggulan yang tidak tersedia pada teknologi identifikasi lain. RFID dapat disediakan dalam bentuk *tag* yang hanya dapat dibaca saja (*read only*) atau dapat dibaca dan ditulis (*read* atau *write*), tidak memerlukan kontak langsung maupun jalur cahaya untuk dapat beroperasi, dapat berfungsi pada berbagai variasi kondisi lingkungan, dan menyediakan tingkat integritas data yang tinggi. Sebagai tambahan, karena teknologi ini sulit untuk dipalsukan, maka RFID dapat menyediakan tingkat keamanan yang tinggi.

2.4.1 RFID Reader

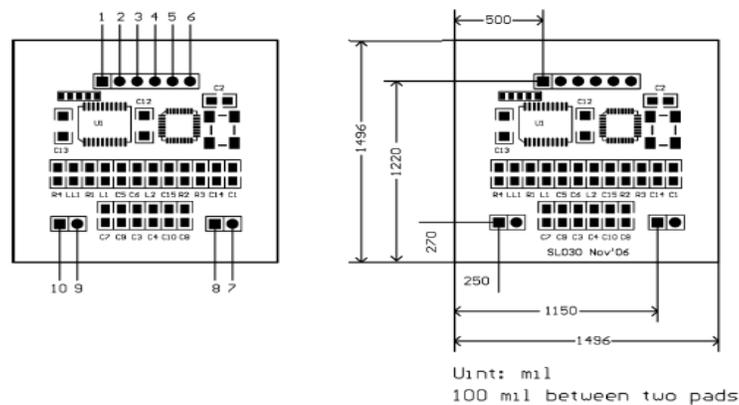
Sebuah *Reader* RFID adalah benar-benar radio, seperti yang ada di dalam mobil, kecuali *Reader* RFID mengambil menggunakan sinyal analog bukan *hip-hop*. *Reader*-nya menghasilkan listrik yang mengalir melalui kabel, listrik tersebut mengenai sebagian logam pada antena dan mengeluarkan sinyal dengan nilai frekuensi dan panjang gelombang tertentu.

Reader-nya tidak hanya menghasilkan sinyal yang dikirim melalui antena ke angkasa, tetapi juga mendengarkan respon dari *tag*. *Reader* RFID seperti sebuah mesin kode *morse* dengan teknologi tinggi, tetapi sebagai ganti dari titik (.) dan *dash* (-) yang didengarkan oleh seorang polisi hutan, *Reader* RFID mengirimkan dan menerima gelombang analog dan mengubah mereka menjadi untaian nol dan satu, bit dari informasi digital. Setiap *reader* terhubung dengan satu atau lebih antena.

Modul *reader* RFID yang digunakan pada penelitian adalah modul SL030 yang merupakan modul *reader* RFID produksi StrongLink. Fitur-fitur dari SL030 sendiri dapat dilihat di bawah ini.

- *Tag* yang dapat dibaca: *Mifare* 1K, *Mifare* 4K, *Mifare* UltraLight
- Mendeteksi *tag* secara otomatis
- Antena terintegrasi
- Komunikasi I²C dengan frekuensi 0 - 400 KHz
- Tegangan operasi DC2.5V - DC3.6V, I/O pins dibolehkan sampai 5V
- Arus pada saat SL030 bekerja 40mA @3.3V
- Arus saat *stand by* 10uA

- Jarak operasi: *Up to 50mm, depending on tag*
- Temperatur penyimpanan: $-40\text{ }^{\circ}\text{C} \sim +85\text{ }^{\circ}\text{C}$
- Temperatur operasi: $-25\text{ }^{\circ}\text{C} \sim +70\text{ }^{\circ}\text{C}$
- Dimensi: $38 \times 38 \times 3\text{ mm}$
- Pin *OUT* akan *low* pada saat *tag* terdeteksi dan *high* pada saat *tag* tidak terdeteksi.



Gambar 2.22 Modul *Reader* RFID SL030
(SL030 user manual,2007)

Tabel 2.6 Tabel Konfigurasi Pin SL030

PIN	SYMBOL	TYPE	DESCRIPTION
1	VDD	PWR	Power supply, DC2.5V to DC3.6V
2	IN	Input	Falling edge wake up SL030 from power down mode
3	SDA	Input/Output	Serial Data Line
4	SLC	Input	Serial Clock Line
5	Out	Output	Tag detect signal low level indicating tag in high level indicating tag out
6	GND	PWR	Ground
7	NC		
8	NC		
9	NC		
10	NC		

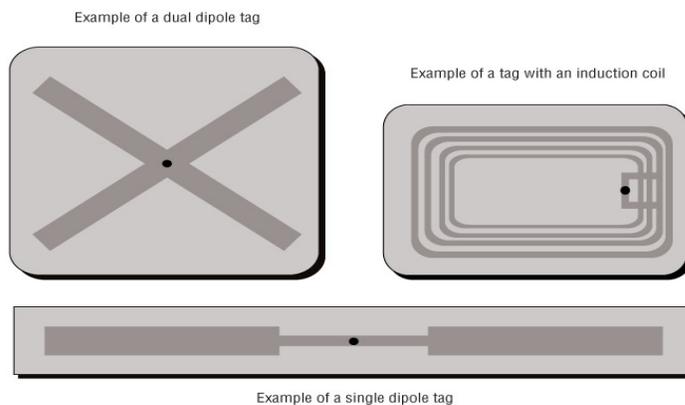
2.4.2 Tag RFID

Tag RFID dibuat dari dua bagian dasar: *chip*, atau *integrated circuit* (IC), dan antena. *Chip* adalah sebuah *computer* kecil yang menyimpan nomor seri unik

dari *chip*. *Chip* juga memiliki *logic* untuk memberitahu dirinya apa yang harus dilakukan saat berada didepan *reader*. Antena memungkinkan *chip* untuk menerima tenaga dan berkomunikasi, memungkinkan *tag* RFID untuk bertukar data dengan *reader*.

Beberapa *tag* adalah *tag* aktif karena mendapat tenaga baterai untuk komunikasi. Sebagian besar *tag* yang diproduksi saat ini adalah *tag* pasif. Ini berarti saat mereka berkomunikasi adalah saat berdekatan dengan *reader*. Menjadi *reader* saat ini berarti mereka berada dalam medan elektromagnetik. Saat *tag* pasif memasuki medan listrik atau *magnetic*, *tag* mendapatkan energi yang cukup dari medan tersebut untuk memberikan tenaga pada dirinya dan menyiarkan informasinya.

Tipe komunikasi yang mengijinkan perpindahan ini terjadi disebut *backscatter*. *Reader* mengirimkan gelombang elektromagnetik pada satu frekuensi yang spesifik. Gelombang itu mengenai *Tag* RFID, dan *tag* kemudian *Scatters back* gelombangnya dengan frekuensi berbeda dengan menyandikan informasi dari *chip* pada gelombang *backscatter* tersebut.



Gambar 2.23 *Tag* RFID

2.4.3 ISO/IEC JT1/SC17

International Organization for Standardization (ISO) telah bekerjasama dengan *International Electrotechnical Commission* (IEC) untuk membuat standar untuk mengidentifikasi kartu dan alat yang berhubungan.

Tabel 2.7 Tabel ISO/IEC Standar Identifikasi Kartu

Standard	Tipe dari ID Card	Hal yang dibahas
ISO/IEC 10536 mengidentifikasi kartu — <i>contactless</i> <i>integrated</i> <i>circuit(s) cards</i>	<i>Smart identification</i> <i>cards</i> , menggunakan RFID pada 13.56 MHz	Part 1: karakter fisik Part 2: dimensi dan letak dari <i>area coupling</i> Part 3: Sinyal elektronik dan prosedur <i>reset</i> Part 4: Protokol untuk menjawab <i>reset</i> dan kiriman
ISO/IEC 14443 mengidentifikasi kartu — <i>Proximity</i> <i>integrated</i> <i>circuit(s) cards</i>	<i>Smart identification</i> <i>cards</i> dengan jarak panjang (hingga 1 meter), menggunakan RFID pada 13.56 MHz	Part 1: karakter fisik Part 2: <i>Interface</i> udara Part 3: Inisialisasi dan <i>anticollision</i> Part 4: <i>Protocol</i> pengiriman
ISO/IEC 15693 <i>Contactless</i> <i>Integrated</i> <i>circuit(s) cards</i> — <i>Vicinity cards</i>		Part 1: karakter fisik Part 2: <i>Interface</i> udara dan inisialisasi Part 3: <i>Protocol anticollison</i> dan pengiriman

ISO 10536, 14443, dan 15693 mencakup karakter fisik, *interface* udara dan *inisialisasi*, dan *protocol anticollision* dan pengiriman dari *Vicinity cards* (*contactless Intergrated circuit cards*, juga dikenal sebagai *Smart Identification Cards*). Mereka juga digunakan pada *proximity cards*, cakupan area seperti tenaga frekuensi radio dan *interface signal*. *Tag* pada *smart card* ini dapat digunakan pada beberapa aplikasi sehingga pada satu kartu yang sama dapat digunakan untuk akses masuk, *login computer*, dan pembayaran kantin.

2.4.4 Frekuensi Kerja RFID

Faktor penting yang harus diperhatikan dalam RFID adalah frekuensi kerja dari sistem RFID. Frekuensi ini digunakan untuk komunikasi *wireless* antara *reader* dengan *tag*. Pemilihan frekuensi kerja berpengaruh pada jarak komunikasi, interferensi dengan sistem radio lain, kecepatan baca, dan ukuran antena. Pada frekuensi rendah pada umumnya digunakan *tag* pasif sementara pada frekuensi yang tinggi digunakan *tag* aktif.

Pada frekuensi rendah, *tag* pasif tidak dapat mentransmisikan data dengan jarak yang jauh, karena keterbatasan daya yang diperoleh dari medan elektromagnetik. Akan tetapi komunikasi tetap dapat dilakukan tanpa kontak langsung. Pada kasus ini hal yang perlu diperhatikan adalah *tag* pasif harus terletak jauh dari objek logam, karena logam secara signifikan mengurangi *fluks* dari medan magnet. Akibatnya *tag* RFID tidak bekerja dengan baik, karena *tag* tidak menerima daya minimum untuk dapat bekerja.

Frekuensi komunikasi yang digunakan tergantung pada suatu aplikasi, dan memiliki rentang dari 125KHz sampai 2,45GHz. Pada frekuensi tinggi juga

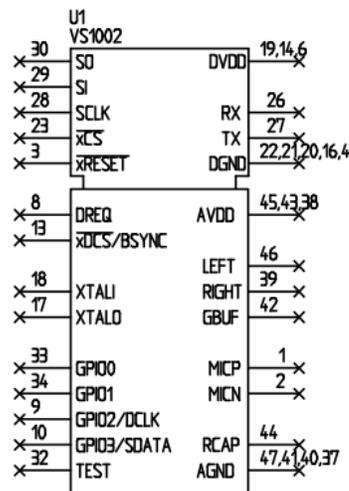
mendapatkan pelemahan ketika *tag* tertutupi oleh es atau air. Pada kondisi terburuk, *tag* yang tertutup oleh logam tidak terdeteksi oleh pembaca RFID. (<http://subari.blogspot.com/2008/02/rfid-radio-frequency-identification.html>)

Ada empat macam RFID *tag* yang sering digunakan bila di dikategorikan berdasarkan frekuensi radio, yaitu :

- *Low frequency tag* (antara 125 sampai 134 KHz)
- *High frequency tag* (13.56 MHz)
- *Ultra high frequency tag* (868 sampai 956 MHz)
- *Microwave tag* (2.45 GHz)

2.5 MP3 Decoder VS1002

MP3 *decoder* yang digunakan pada penelitian adalah VS1002 dari *VLSI Solution* yang merupakan perusahaan pembuat IC yang bertempat di Tampere, Finlandia. VS1002 dapat men-*decode* MPEG 1.0 dan 2.0 *audio layer 3*, WAV, dan *file* PCM.



Gambar 2.24 Konfigurasi Pin VS1002

Pada saat awal (*boot*) apabila xRESET diberi logika *high* dengan menggunakan resistor *pull up* maka VS1002 akan berganti mode dari *normal mode* menjadi *SPI boot mode*. Perubahan yang terjadi dari *normal mode* ke *SPI boot mode* dapat dilihat pada tabel 2.8.

Tabel 2.8 Tabel Perubahan Fungsi Pin Dari *Normal Mode* Ke *SPI Boot Mode*

Normal Mode	SPI Boot Mode
GPIO0	xCS
GPIO1	CLK
DREQ	MOSI
GPIO2	MISO

Pada *SPI boot mode* VS1002 akan mencoba mengakses memori eksternal dengan komunikasi SPI melalui GPIO0, GPIO1, DREQ, dan GPIO2 yang telah berubah fungsi. Pada saat *normal mode* VS1002 pun memiliki *port* SPI, yaitu pada pin SO, SI, SCLK, XCS dan XDCS, tetapi *port* SPI ini tidak dapat mengakses langsung memori eksternal seperti pada *SPI boot mode* di atas. (Datasheet VLSI VS1002d, 2005)

2.5.1 SPI Bus

SPI *bus* yang digunakan pada *Serial Data Interface* (SDI) dan *Serial Command Interface* (SCI) VS1002 merupakan SPI *bus* yang sama dengan yang dipakai oleh Motorola. VS1002 memiliki 2 mode komunikasi SPI, yaitu *native mode* dan *compatibility mode*. Pada *native mode* pin DCLK, SDATA, dan BSYNC akan diganti secara berurutan dengan GPIO2, GPIO3, dan XDCS.

Tabel 2.9 Tabel Fungsi Pin Pada Komunikasi SDI Dan SCI Dengan *Native Mode*

SDI Pin	SCI Pin	Keterangan
XDCS	XCS	<i>Active low chip select input.</i> Kondisi <i>high</i> pada pin ini akan mengakibatkan <i>serial interface</i> masuk ke <i>standby mode</i> dan mengakhiri operasi yang sedang berjalan. Kondisi <i>high</i> juga akan mengakibatkan <i>serial output (SO)</i> masuk ke kondisi <i>high impedance</i> . Bila <i>SM_SDISHARE</i> bernilai 1 maka pin XDCS tidak digunakan, tetapi sinyal XDCS akan dihasilkan secara internal dengan meng- <i>invert</i> pin XCS.
SCK		<i>Serial clock input.</i> SCK juga digunakan sebagai <i>master clock</i> di dalam VS1002 untuk <i>register interface</i> .
SI		<i>Serial input.</i> Apabila <i>chip select</i> aktif maka data akan diambil pada saat <i>SCK rising edge</i> .
-	SO	<i>Serial output.</i> Data akan keluar pada saat <i>SCK falling edge</i> .

Pada *compatibility mode* DCLK, SDATA, dan BSYNC akan aktif. (Datasheet VLSI VS1002d, 2005)

Tabel 2.10 Tabel Fungsi Pin Pada Komunikasi SDI Dan SCI Dengan *Compatibility Mode*

SDI Pin	SCI Pin	Keterangan
-	XCS	<p><i>Active low chip select input.</i></p> <p>Kondisi <i>high</i> pada pin ini akan mengakibatkan <i>serial interface</i> masuk ke <i>standby mode</i> dan mengakhiri operasi yang sedang berjalan. Kondisi <i>high</i> juga akan mengakibatkan <i>serial output (SO)</i> masuk ke kondisi <i>high impedance</i>. SDI akan selalu aktif.</p>
BSYNC	-	Data SDI disinkronisasi pada saat BSYNC <i>rising edge</i> .
DCLK	SCK	<p><i>Serial Clock Input</i></p> <p>Serial <i>clock</i> juga digunakan secara internal sebagai <i>master clock</i> untuk <i>interface register</i>.</p> <p><i>Rising edge</i> yang pertama setelah XCS bernilai 0 akan menandakan bit pertama yang akan ditulis.</p>
SDATA	SI	<p><i>Serial Input</i></p> <p>Data pada SI akan diambil pada saat SCK <i>rising edge</i> dan XCS bernilai 0.</p>
-	SO	<p><i>Serial Output</i></p> <p>Pada saat proses membaca SO akan mengeluarkan data pada saat SCK <i>falling edge</i> dan pada saat proses menulis SO akan berada pada kondisi <i>high impedance</i>.</p>

Compatibility mode ini merupakan mode SPI yang digunakan pada VS1001, sementara *native mode* merupakan mode baru yang dikembangkan *VLSI Solution* untuk VS1002. Pada penelitian dibuat dengan menggunakan *native mode*. Mode yang aktif dapat dipilih dengan mengubah nilai *register* MODE pada *register* SCI. Penjelasan lebih lanjut dapat dilihat pada tabel 2.11 dan tabel 2.12.

A. Protokol Serial Untuk *Serial Data Interface*

Pada *native mode* sinkronisasi byte didapatkan dari XDCS, sehingga XDCS tidak boleh berubah nilainya pada saat transfer data berlangsung. Protokol *serial data interface* digunakan juga untuk mengirim data .MP3 yang ingin di *decode* ke dalam VS1002.

B. Protokol Serial Untuk *Serial Command Interface*

Protokol serial untuk *serial command interface* terdiri dari 1 *Byte* instruksi, 1 *address Byte*, dan 16-bit data. Setiap operasi *read* atau *write* dapat menulis atau membaca 1 *register*. Data per bit dibaca pada saat *rising edge*, sehingga *user* harus meng-*update* data pada saat *falling edge*. Seluruh operasi dari *serial command interface* pada penelitian dilakukan dengan mengirim MSB (*Most Significant Bit*) terlebih dahulu.

Pada saat menjalankan SCI *read* atau *write*, XCS harus diset 0. Nilai *Byte* instruksi untuk operasi *read* adalah 0x03, sedangkan nilai *Byte* instruksi untuk operasi *write* adalah 0x02.

C. Register Serial Command Interface

Register SCI merupakan *register* yang berisi instruksi-instruksi untuk mengontrol operasi dari VS1002. Fungsi utama dari *command interface* adalah:

- Mengontrol mode operasi
- Meng-*upload user program*
- Mengakses data *header*
- Informasi status
- Mengakses data digital yang telah di-*encode*

Panjang data yang dikirim melalui SCI selalu 16 bits. VS1002 dikontrol dengan membaca dan menulis SCI *register*. Tabel 2.11 merupakan tabel *register* SCI.

Tabel 2.11 Tabel keterangan *register* SCI

SCI registers, prefix SCL, offset 0xC000				
Reg	Type	Reset	Abbrev[bits]	Description
0x0	rw	0x800	MODE	Mode control.
0x1	rw	0x2C ¹	STATUS	Status of VS1002d.
0x2	rw	0	BASS	Built-in bass enhancer.
0x3	rw	0	CLOCKF	Clock freq + doubler.
0x4	r	0	DECODE_TIME	Decode time in seconds.
0x5	rw	0	AUDATA	Misc. audio data.
0x6	rw	0	WRAM	RAM write.
0x7	rw	0	WRAMADDR	Base address for RAM write.
0x8	r	0	HDATA0	Stream header data 0.
0x9	r	0	HDATA1	Stream header data 1.
0xA	rw	0	AIADDR	Start address of application.
0xB	rw	0	VOL	Volume control.
0xC	rw	0	AICTRL0	Application control register 0.
0xD	rw	0	AICTRL1	Application control register 1.
0xE	rw	0	AICTRL2	Application control register 2.
0xF	rw	0	AICTRL3	Application control register 3.

Tabel 2.12 Tabel Keterangan *Register* MODE

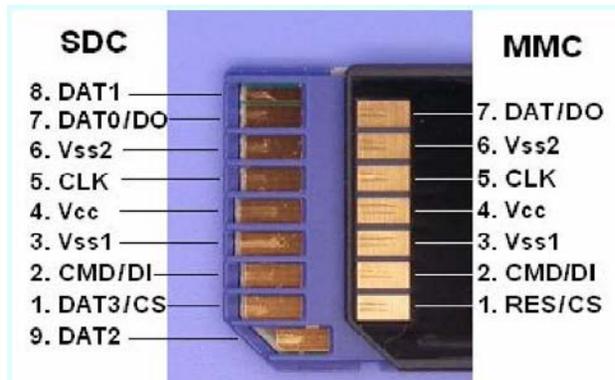
Bit	Name	Function	Value	Description
0	SM_DIFF	Differential	0 1	normal in-phase audio left channel inverted
1	SM_SETTOZERO	Set to zero	0 1	right wrong
2	SM_RESET	Soft reset	0 1	no reset reset
3	SM_OUTOFWAV	Jump out of WAV decoding	0 1	no yes
4	SM_PDOWN	Powerdown	0 1	power on powerdown
5	SM_TESTS	Allow SDI tests	0 1	not allowed allowed
6	SM_STREAM	Stream mode	0 1	no yes
7	SM_PLUSV	MP3+V active	0 1	no yes
8	SM_DACT	DCLK active edge	0 1	rising falling
9	SM_SDIORD	SDI bit order	0 1	MSb first MSb last
10	SM_SDISHARE	Share SPI chip select	0 1	no yes
11	SM_SDINEW	VS1002 native SPI modes	0 1	no yes
12	SM_ADPCM	ADPCM recording active	0 1	no yes
13	SM_ADPCM_HP	ADPCM high-pass filter active	0 1	no yes

2.6 Multi Media Card (MMC) Atau Secure Digital Card (SD Card)

MMC atau SD *card* merupakan media penyimpanan data yang biasa digunakan pada *portable device*. SD *card* merupakan pengembangan dari MMC. Tidak banyak perbedaan antara SD *card* dengan MMC, di antaranya adalah:

- Ukuran SD *card* lebih tebal dari MMC
- SD *card* memiliki *switch* untuk *write protection*, sedangkan MMC tidak

- *SD card* memiliki 9 pin, sedangkan MMC memiliki 7 pin. Susunan pin dari *SD card* dan MMC dapat dilihat pada gambar 2.25.



Gambar 2.25 Penomeran Pin Pada *SD Card* Dan MMC

- *SD card* memiliki 3 mode transfer data, yaitu mode SD, mode MMC, dan mode SPI. Sedangkan MMC hanya memiliki 2 mode transfer data, yaitu mode MMC dan mode SPI.
- Kecepatan maksimum transfer data *SD card* adalah 25Mbit/s, lebih cepat dari MMC yang memiliki kecepatan maksimum 20Mbit/s dengan mode komunikasi yang sama yaitu mode SPI atau mode MMC. Sedangkan pada mode SD kecepatan transfer data maksimum pada *SD card* dapat mencapai 100Mbit/s.

Agar *device* dapat meng-*interface* baik MMC maupun *SD card*, maka mode transfer data yang digunakan harus sama, dalam hal ini mode transfer data mode SD yang ada pada *SD card* tidak dapat digunakan, karena pada MMC mode transfer data tersebut tidak ada. Pada penelitian mode transfer data yang digunakan adalah mode SPI. Penjelasan mengenai konfigurasi pin dari masing-masing mode transfer dapat dilihat pada tabel 2.13, tabel 2.14, tabel 2.15, dan tabel 2.16.

Tabel 2.13 Tabel Mode MMC Pada *Multi Media Card* (MMC)

PIN	NAMA	KETERANGAN
1	RSV	NC
2	CMD	Data perintah
3	VSS	<i>Ground</i>
4	VDD	Tegangan <i>supply</i>
5	CLK	<i>Clock</i>
6	VSS	<i>Ground</i>
7	DAT	Data

Keterangan : NC = Not Connected / Tidak dihubungkan

Tabel 2.14 Tabel Mode SPI Pada *Multi Media Card* (MMC)

PIN	NAMA	KETERANGAN
1	CS	<i>Chip Select</i>
2	DI	<i>Data In</i>
3	VSS	<i>Ground</i>
4	VDD	Tegangan <i>supply</i>
5	SCK	<i>Clock</i>
6	VSS	<i>Ground</i>
7	DO	<i>Data Out</i>

Tabel 2.15 Tabel *SD Bus Mode* Pada *SD Card*

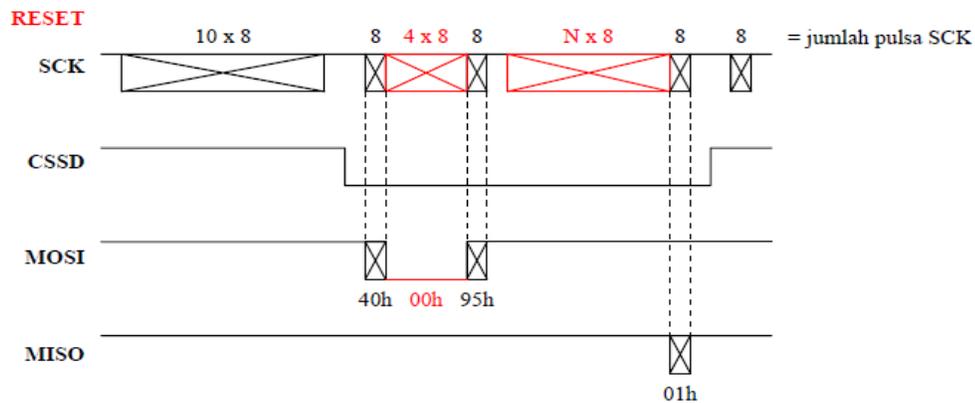
PIN	NAMA	KETERANGAN
1	CD/DAT3	Deteksi kartu / jalur data bit ke 3
2	CMD	Data perintah
3	VSS1	<i>Ground</i>
4	VDD	Tegangan <i>Supply</i>
5	CLK	<i>Clock</i>
6	VSS2	<i>Ground</i>
7	DAT0	Jalur data bit ke 0
8	DAT1	Jalur data bit ke 1
9	DAT2	Jalur data bit ke 2

Tabel 2.16 Tabel Mode *SPI* Pada *SD Card*

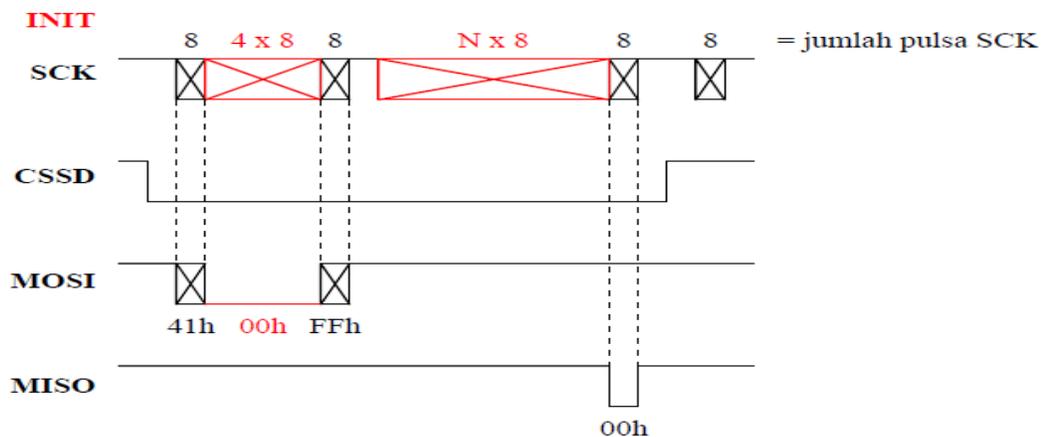
PIN	NAMA	KETERANGAN
1	CS	<i>Chip Select</i>
2	DI	Data perintah
3	VSS1	<i>Ground</i>
4	VDD	Tegangan <i>Supply</i>
5	SCK	<i>Clock</i>
6	VSS2	<i>Ground</i>
7	DO	Jalur data bit ke 0
8	RSV	NC
9	RSV	NC

Pada mode MMC pin 8 dan 9 dari SD *card* tidak digunakan sementara fungsi pin 1 sampai 7 dari SD *card* akan sama dengan fungsi pin 1 sampai 7 pada mode MMC.

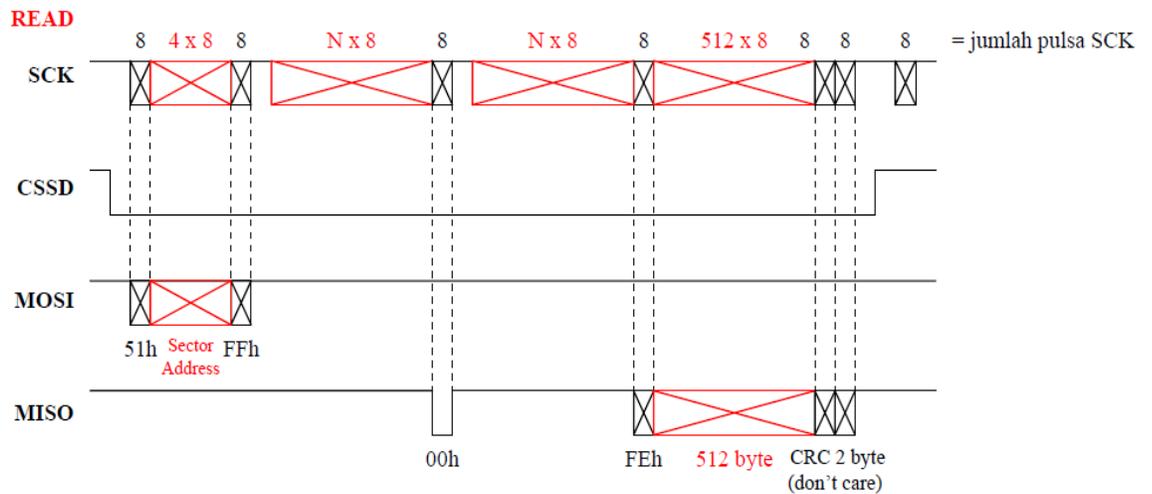
Pada intinya *device* dengan *slot* SD *card* dapat menggunakan MMC atau SD *card*, sedangkan *device* dengan *slot* MMC tidak dapat menggunakan SD *card* karena ukuran SD *card* yang lebih tebal sehingga SD *card* tidak dapat masuk ke *slot* MMC. Mode transfer data yang digunakan pada penelitian adalah SPI mode. Sedangkan format MMC/SD *card* yang digunakan adalah FAT16. Pada gambar 2.26 merupakan *timing diagram* SD *card* reset, pada gambar 2.27 merupakan *timing diagram* SD *card* init, dan pada gambar 2.28 merupakan *timing diagram* SD *card* read.



Gambar 2.26 *Timing Diagram* SD Card Reset



Gambar 2.27 *Timing Diagram* SD Card Init



Gambar 2.28 *Timing Diagram SD Card Read*

2.6.1 File Allocation Table 16 (FAT16)

File Allocation Table (FAT) adalah sebuah metode untuk melakukan alokasi *file* atau data pada suatu memori yang dikembangkan oleh Microsoft. Standar dari MMC atau SD *card* adalah diformat dengan menggunakan format FAT16. Angka 16 pada FAT16 menunjukkan bahwa alokasi data dibuat dengan menggunakan format 16 bit, sehingga jumlah unit alokasi yang dapat dibentuk adalah sejumlah 2^{16} atau 65.536 unit alokasi. Tetapi setelah diformat dengan FAT16 maka akan ada beberapa unit alokasi yang dipesan untuk keperluan tertentu sehingga unit alokasi maksimum yang dapat dicapai apabila sebuah MMC atau SD *card* diformat dengan FAT16 adalah 65.524 buah unit alokasi.

Pada format FAT dikenal istilah *cluster*. *Cluster* merupakan nama untuk unit alokasi yang disebut di atas. Dengan kata lain dapat disebutkan bahwa pada format FAT16 jumlah *cluster* maksimum yang dapat dicapai adalah 65.524 *cluster*. Sebuah *cluster* dibagi menjadi beberapa *sector*. Ukuran dari sebuah *cluster* tergantung dari berapa banyak *sector* di dalamnya. Sebuah *sector* biasanya

memiliki ukuran 512 bytes, terkadang Microsoft membuat ukuran *sector* seolah-olah menjadi 1024 bytes dengan cara menganggap 2 *sector* menjadi 1 *sector*, tetapi pada penelitian ukuran *sector* yang digunakan adalah 512 bytes.

Tabel 2.17 Tabel Ukuran Volume (*Drive*) Yang Dapat Dicapai Berdasarkan Ukuran *Cluster*

Ukuran Volume (<i>Drive</i>)	Ukuran <i>Cluster</i>
17 MB – 32 MB	512 bytes
33 MB – 64 MB	1 KB
65 MB – 128 MB	2 KB
129 MB – 256 MB	4 KB
257 MB – 512 MB	8 KB
512 MB – 1.024 MB (1 GB)	16 KB
1.025 MB – 2 GB	32 KB
2 GB – 4 GB	64 KB

Pada penelitian *boot sector* yang dapat dibaca hanyalah *boot sector* yang berada pada alamat 0. Penjelasan mengenai isi dari *boot sector* dapat dilihat pada tabel 2.18.

Tabel 2.18 Tabel Keterangan *Boot Sector*

Alamat		Ukuran	Keterangan
Heksadesimal	Desimal		
0000h	0	3 byte	Code to jump to the bootstrap code
0003h	3	8 byte	Oem ID – nama sistem operasi yang melakukan proses format
000Bh	11	2 byte	Bytes per Sector – biasanya ada 512 byte tiap sector
000Dh	13	1 byte	Sectors per Cluster
000Eh	14	2 byte	Reserved Sectors from the start of the volume
0010h	16	1 byte	Number of FAT copies – biasanya 2 kopi FAT digunakan untuk mencegah hilangnya data
0011h	17	2 byte	Number of possible root entries – disarankan 512 entri
0013h	19	2 byte	Small number of sectors – digunakan jika kapasitas kurang dari 32 MB
0015h	21	1 byte	Media Descriptor
0016h	22	2 byte	Sectors per FAT
0018h	24	2 byte	Sectors per Track
001Ah	26	2 byte	Number of Heads
001Ch	28	4 byte	Hidden Sectors
0020h	32	4 byte	Large number of sectors – digunakan jika kapasitas lebih besar dari 32 MB
0024h	36	1 byte	Drive Number – digunakan oleh beberapa <i>bootstrap code</i> , fx. MS-DOS
0025h	37	1 byte	Reserved – digunakan Windows NT untuk menentukan apakah integritas media penyimpanan ini harus diperiksa
0026h	38	1 byte	Extended Boot Signature – mengindikasikan bahwa 3 <i>field</i> berikutnya tersedia
0027h	39	4 byte	Volume Serial Number
002Bh	43	11 byte	Volume Label – seharusnya sama dengan yang ada di <i>root directory</i>
0036h	54	8 byte	File System Type – berisi karakter 'FAT16'
003Eh	62	448 byte	Bootstrap code
01FEh	510	2 byte	Boot sector signature - 55h dan AAh

Untuk membaca data pada format FAT16 ada beberapa hal yang harus diperhatikan, yaitu alamat awal FAT, alamat awal *root directory*, dan alamat awal *data region*. Alamat-alamat tersebut dapat dihitung dari data-data yang ada pada *boot sector*. Perhitungan untuk mengetahui alamat-alamat tersebut adalah :

- Alamat awal FAT = Reserved Sectors * 512
- Alamat awal *root directory* = (Sectors per FAT * Number of FAT * 512) +
Alamat awal FAT
- Alamat awal *data region* = (Number of possible root entries * 32) +
Alamat awal *root directory*

Masing-masing *file* pada *memory card* memiliki 32 *Byte entry* pada *root directory*. Maksimum *file* atau *folder* yang ada pada *root directory* adalah 512. Penjelasan mengenai 32 *Byte entry* tersebut dapat dilihat pada tabel 2.19.

Tabel 2.19 Tabel Keterangan 32 *Byte entry* Untuk 1 *File*

Alamat		Ukuran	Keterangan
Heksadesimal	Desimal		
00h	0	8 byte	File Name – Nama file
08h	8	3 byte	File Name Extension – Nama ekstensi file
0Bh	11	1 byte	Attribute Byte – Byte atribut
0Ch	12	1 byte	Reserved for Windows NT – bernilai 00h
0Dh	13	1 byte	Creation Milisecond Stamp – Waktu pembuatan file (seperseratus detik), bernilai 0 hingga 199
0Eh	14	2 byte	Creation Time – Waktu pembuatan file
10h	16	2 byte	Creation Date – Tanggal pembuatan file
12h	18	2 byte	Last Access Date – Tanggal akses terakhir
14h	20	2 byte	Reserved for FAT32 – bernilai 0000h untuk FAT16
16h	22	2 byte	Last Write Time – Waktu penulisan file terakhir
18h	24	2 byte	Last Write Date – Tanggal penulisan file terakhir
1Ah	26	2 byte	Starting Cluster – Cluster Awal
1Ch	28	4 byte	File Size – Ukuran file dalam satuan byte

2.7 MP3 (MPEG Audio Layer 3)

MPEG (*Moving Picture Experts Group*) adalah sebuah nama dari standar kompresi untuk *file audio* dan *visual*. *File* MPEG terdiri dari beberapa *frame*. Masing-masing *frame* memiliki *header* pada awal *frame*. *Sample rate* pada masing-masing *frame* bernilai konstan. Ada tiga macam *layer* pada *file* MPEG, yaitu *layer I*, *layer II*, dan *layer III*, masing-masing *layer* tersebut memiliki cara tersendiri untuk melakukan kompresi data *audio* pada *frame*, tetapi ketiga macam *layer* di atas memiliki format *header* yang sama.

Ada beberapa versi kompresi *audio* pada standar MPEG, yaitu MPEG1.0, MPEG2.0, dan MPEG2.5. MPEG2.0 dan MPEG2.5 juga biasa disebut LSF (*Lower*

Sampling Frequency). Pada penelitian versi MPEG yang digunakan adalah MPEG1.0 *layer III*. Sebuah *file audio* dapat dikompresi dengan menggunakan konstan *bitrate* atau *bitrate* yang bervariasi. Dengan kata lain masing-masing *frame* dapat memiliki *bitrate* yang berbeda.

2.7.1 MPEG Audio Frame Header

32 bits pertama pada sebuah *frame* merupakan *header* dari *frame* tersebut. Bit ke 0 dari penjelasan *frame header* pada tabel 2.20 merupakan *most significant bit* (MSB).

Tabel 2.20 Tabel Penjelasan *Frame Header*

Posisi	Lebar Data	Keterangan
0	11	<i>Frame Sync</i> : digunakan untuk mencari <i>header frame</i>
11	2	ID versi <i>audio</i> 00 --- MPEG versi 2.5 01 --- <i>reserved</i> 10 --- MPEG versi 2.0 11 --- MPEG versi 1.0
13	2	Indeks <i>layer</i> 00 --- <i>reserved</i> 01 --- <i>Layer III</i> 10 --- <i>Layer II</i> 11 --- <i>Layer I</i>
15	1	Bit proteksi 0 --- Diproteksi oleh 16 bits CRC setelah <i>header</i>

		1 --- Tidak diproteksi
16	4	Indeks <i>bitrate</i> (lihat tabel 2.21)
20	2	Indeks <i>sample rate</i> (lihat tabel 2.22)
22	1	<i>Padding bit</i> Bila bernilai 1 maka data diblok ke dalam satu <i>slot</i> (penting untuk menghitung ukuran <i>frame</i>).
23	1	<i>Private bit</i>
24	2	<i>Channel mode</i> 00 --- <i>Stereo</i> 01 --- <i>Joint Stereo (Stereo)</i> 10 --- <i>Dual channel (2 channel mono)</i> 11 --- <i>Single Channel (Mono)</i> <i>Dual channel</i> merupakan 2 <i>channel mono</i> yang tidak saling bergantung, masing-masing akan menggunakan setengah <i>bitrate</i> dari <i>file audio</i> .
26	2	Mode <i>extension</i> (hanya pada <i>joint stereo</i>)
28	1	<i>Copyright bit</i>
29	1	<i>Original bit</i>
30	2	<i>Emphasis</i> 00 --- <i>None</i> 01 --- 50/15 ms 10 --- <i>reserved</i>

		<p>11 --- CCIT J.17</p> <p><i>Emphasis</i> memberikan informasi kepada <i>decoder</i> apakah <i>decoder</i> perlu meratakan kembali <i>file audio</i> setelah terkena efek seperti <i>Dolby</i> (jarang digunakan).</p>
--	--	---

Tabel 2.21 Tabel Indeks *Bitrate* (*Bitrate* Dalam Kbps)

Indeks <i>Bitrate</i>	MPEG1.0			MPEG2.0 & 2.5	
	<i>Layer I</i>	<i>Layer II</i>	<i>Layer III</i>	<i>Layer I</i>	<i>Layer II & III</i>
0000	<i>free</i>				
0001	32	32	32	32	8
0010	64	48	40	48	16
0011	96	56	48	56	24
0100	128	64	56	64	32
0101	160	80	64	80	40
0110	192	96	80	96	48
0111	224	112	96	112	56
1000	256	128	112	128	64
1001	288	160	128	144	80
1010	320	192	160	160	96
1011	352	224	192	176	112
1100	384	256	224	192	128
1101	416	320	256	224	144

1110	448	384	320	256	160
1111	<i>reserved</i>				

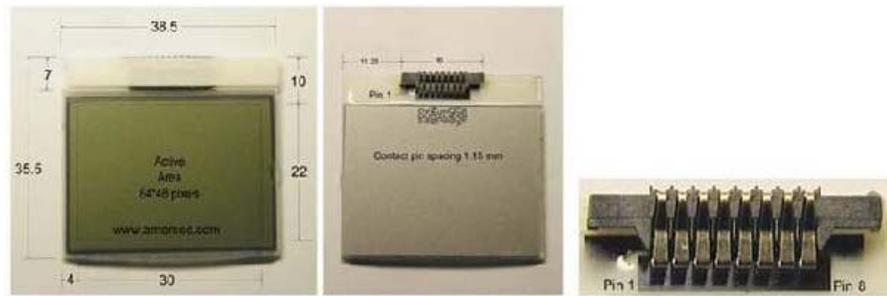
Keterangan: *free = file* menggunakan *bitrate* konstan, di mana *bitrate* tidak ditentukan sebelumnya. Hanya sedikit *decoder* yang dapat meng-*encode file* seperti ini.

Tabel 2.22 Tabel Indeks *Sample Rate*

Indeks <i>Sample Rate</i>	MPEG1.0	MPEG2.0	MPEG2.5
00	44100 Hz	22050 Hz	11025 Hz
01	48000 Hz	24000 Hz	12000 Hz
10	32000 Hz	16000 Hz	8000 Hz
11	<i>reserved</i>		

2.8 LCD LPH 7779

LCD grafik LPH 7779 merupakan sebuah LCD grafik satu warna (*monochrome*) yang memiliki resolusi sebesar 48 x 84 *pixel*. LCD jenis ini dapat di temukan pada layar *handphone* produksi nokia seperti tipe 3310. Pada LCD ini didalamnya telah terdapat sebuah IC sebagai *driver* atau *controller* 48 x 84 *pixel* yang merupakan produksi dari Philips yaitu PCD 8544 dengan penerimaan data secara serial. Panjang dan lebar *display* adalah 38.5 x 35.5 mm dengan panjang dan lebar *display* yang aktif adalah 30 x 22 mm.



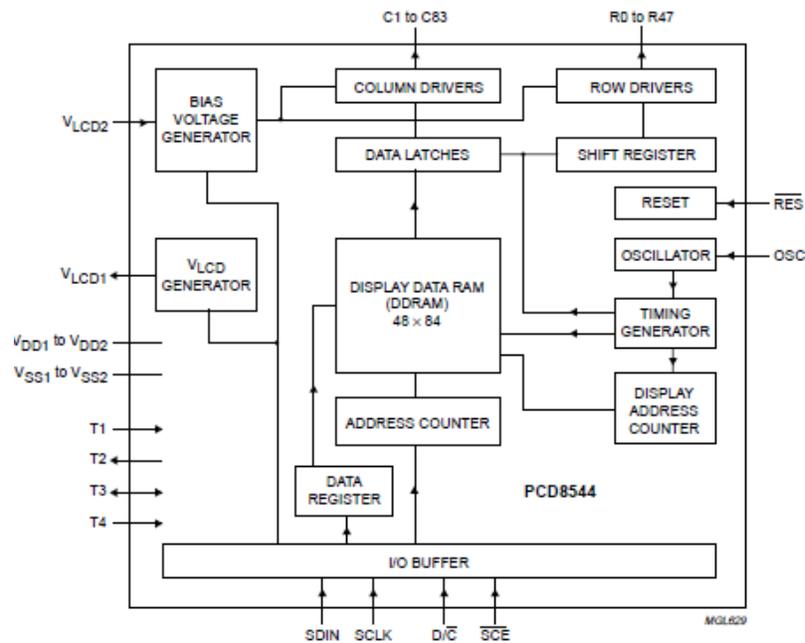
Gambar 2.29 LCD Grafik LPH 7779
http://www.amontec.com/lcd_nokia_3310.shtml

Tabel 2.23 Tabel Keterangan Pin LPH 7779

Pin	Sinyal	Deskripsi
1	Vdd	Pin untuk <i>supply</i>
2	SCLK	Pin untuk <i>input</i> sinyal <i>clock</i>
3	SDIN	Pin untuk <i>input</i> data serial
4	D/C	Pin mode <i>select</i> (data atau <i>command</i>)
5	SCE	Pin <i>input chip enable</i>
6	<i>Ground</i>	Pin <i>ground</i>
7	<i>Vout</i>	Pin <i>output</i>
8	RES	Pin untuk <i>reset</i>

2.9 PCD 8544

IC PCD 8544 merupakan sebuah IC *driver* atau *controller* dengan teknologi CMOS yang digunakan untuk men-*drive* LCD grafik dengan resolusi 48 x 84 *pixel*. PCD 8544 diakses dengan komunikasi serial melalui mikrokontroler dan semua fungsi fungsi yang di perlukan untuk LCD (*display*) telah tersedia dalam IC ini.



Gambar 2.30 Diagram Blok PCD 8544
(Datasheet Philips Semiconductor, PCD 8544, 1999)

2.9.1 Deskripsi Pin PCD 8544

Tabel 2.24 Tabel Deskripsi Pin PCD 8544

Simbol	Deskripsi
R0-R47	<i>Output driver baris LCD</i>
C0-C83	<i>Output driver kolom LCD</i>
V_{SS1}, V_{SS2}	<i>Ground</i>
V_{DD1}, V_{DD2}	<i>Tegangan supply</i>
V_{LCD1}, V_{LCD2}	<i>Tegangan supply LCD</i>
T1	<i>Input test 1</i>
T2	<i>Input test 2</i>
T3	<i>Input test 3</i>
T4	<i>Input test 4</i>

SDIN	<i>Input data serial</i>
SCLK	<i>Input clock serial</i>
D/C	<i>Mode select (data atau command)</i>
SCE	<i>Chip enable</i>
OSC	osilator
RES	<i>Reset</i>
Dummy 1,2,3	NC

2.9.2 Fitur-fitur PCD 8544

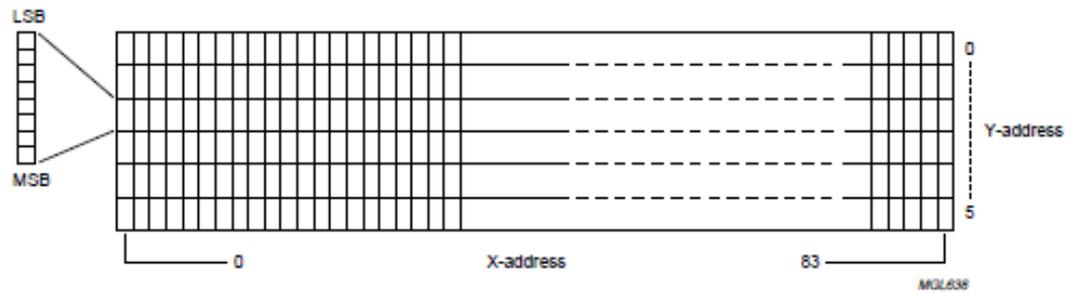
- Merupakan *single chip LCD controller / driver*
- Memiliki *output* 48 baris dan 84 kolom
- DDRAM 48 x 84 bit
- *On chip* :
 - Pembangkit tegangan *supply* LCD (dapat juga di *supply* secara eksternal)
 - Pembangkit tegangan bias LCD
 - Tidak memerlukan komponen osilator eskternal (dapat juga diberikan *clock* secara eksternal)
- Memiliki pin *input reset* eksternal (*active low*)
- *Serial interface* maksimal 4.0 Mbits/s
- *Mux rate* 48
- Tegangan *supply logic* (V_{DD} to V_{SS}) memiliki *range* : 2.7 sampai 3.3 V
- Tegangan *supply* LCD (V_{LCD} to V_{SS}) memiliki *range*

- 6.0 sampai 8.5 V bila tegangan LCD dibangkitkan secara internal
- 6.0 sampai 9 V bila tegangan LCD di-*supply* secara eksternal
- Konsumsi daya rendah
- Memiliki temperatur kompensasi dari LCD
- Temperature *range* :-25 sampai 70 °C

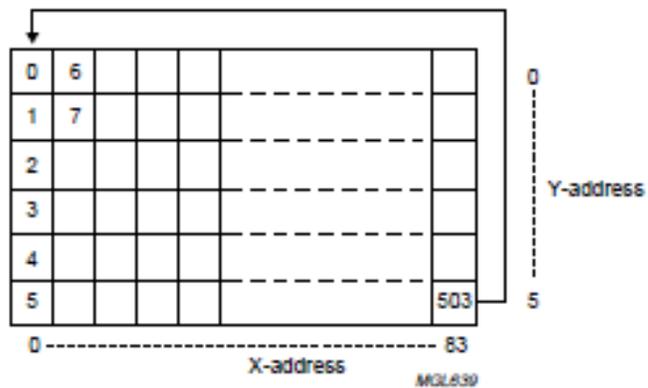
2.9.3 Pengalamatan

Pada PCD 8544 data dimasukan kedalam 48 x 84 bit RAM berupa data byte. Kolom dialamati dengan menggunakan *address pointer*. *Range* dari alamat X adalah mulai 0 sampai 83 (1010011) sedangkan alamat Y dimulai dari 0 sampai 5 (101). Pengalamatan di luar *range* tersebut tidak dikenali oleh PCD 8544.

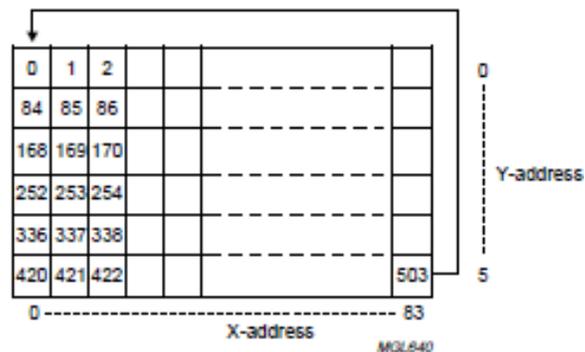
PCD 8544 memiliki dua mode pengalamatan, yaitu secara horizontal dan vertikal. Pada mode pengalamatan secara vertikal ($V=1$), alamat Y di *increment* setelah tiap byte. Setelah alamat Y yang terakhir ($Y=5$) alamat Y akan dimulai lagi dari 0 dan X akan di *increment* ke alamat X pada kolom selanjutnya. Pada mode pengalamatan horizontal ($V=0$), alamat X di *increment* tiap byte. Setelah alamat X yang terakhir ($X=83$), alamat X kembali pada 0 dan Y di *increment* pada alamat baris selanjutnya. Setelah alamat terakhir tercapai ($X=83$ dan $Y=5$), *address pointer* kembali lagi pada $X=0$ dan $Y=0$. Gambar 2.31 menunjukkan format RAM pada PCD 8544, gambar 2.33 menunjukkan mode pengalamatan horizontal dan gambar 2.32 vertikal.



Gambar 2.31 RAM Format Addressing
(Datasheet Philips Semiconductor, PCD 8544, 1999)



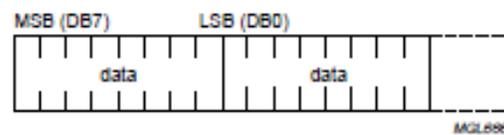
Gambar 2.32 Sekuen Penulisan Byte Data Ke Dalam RAM Dengan Pengalamatan Vertikal (V=1)
(Datasheet Philips Semiconductor, PCD 8544, 1999)



Gambar 2.33 Sekuen Penulisan Byte Data Ke Dalam RAM Dengan Pengalamatan Horizontal (V=0)
(Datasheet Philips Semiconductor, PCD 8544, 1999)

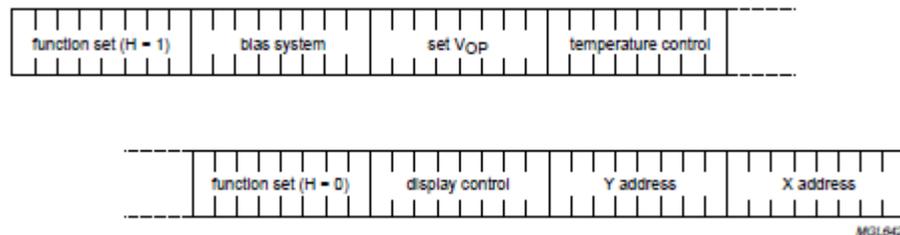
2.9.4 Instruksi

Format instruksi dibagi menjadi 2 mode yaitu *data* dan *command*. Bila pin D/C di set 0 maka (byte yang dikirimkan) akan diterima sebagai *command*. Bila pin D/C di set 1 maka (byte yang dikirimkan) akan diterima sebagai data dan data tersebut akan disimpan pada DRAM. Level dari D/C dibaca setiap bit terakhir dari byte data yang dikirimkan.



Gambar 2.34 Format Umum Aliran Data Serial
(Datasheet Philips Semiconductor, PCD 8544, 1999)

Pada pengiriman data secara serial, bit MSB dikirimkan terlebih dahulu kemudian bit selanjutnya sesuai dengan urutannya. Ketika pin SCE dalam keadaan *high*, sinyal *clock* SCLK diabaikan dan *serial interface* diinisialisasi. Kemudian pada saat SCE *low*, *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 2.35 Contoh Pengiriman Data Serial
(Datasheet Philips Semiconductor, PCD 8544, 1999)

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

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

Tabel 2.25 Tabel Instruksi IC PCD 8544

INSTRUCTION	D/ \bar{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	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	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀	writes data to display RAM
(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
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	sets Y-address of RAM; 0 ≤ Y ≤ 5
Set X address of RAM	0	1	X ₆	X ₅	X ₄	X ₃	X ₂	X ₁	X ₀	sets X-address part of RAM; 0 ≤ X ≤ 83
(H = 1)										
Reserved	0	0	0	0	0	0	0	0	1	do not use
Reserved	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

Tabel 2.26 Tabel Penjelasan Simbol Pada Tabel Instruksi IC PCD 8544

BIT	0	1
PD	chip is active	chip is in Power-down mode
V	horizontal addressing	vertical addressing
H	use basic instruction set	use extended instruction set
D and E	display blank normal mode all display segments on inverse video mode	
TC ₁ and TC ₀	V _{LCD} temperature coefficient 0 V _{LCD} temperature coefficient 1 V _{LCD} temperature coefficient 2 V _{LCD} temperature coefficient 3	