

BAB II

TINJAUAN PUSTAKA

2.1 Sejarah Arah Kiblat

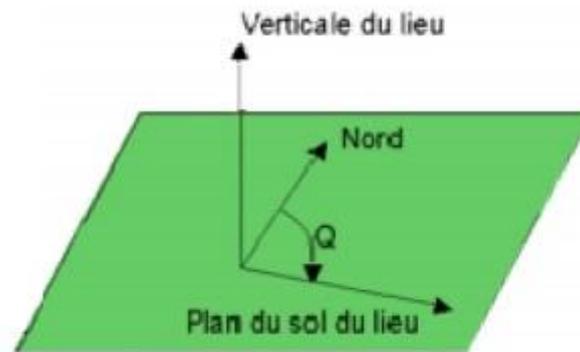
Kiblat adalah kata Arab yang menunjukkan arah yang dituju saat seorang muslim mendirikan sholat. Secara literal kiblat dalam bahasa Arab adalah pemusatan perhatian. Awalnya sebelum ada kiblat, umat Islam awal sholat menghadap ke mana saja. Sehingga, pada tempat yang sama, bisa ada yang menghadap ke timur, barat, atau arah lain sesuka mereka. Kemudian, ditetapkanlah kiblat mengarah ke Masjidil Aqsha di Yerusalem. Menurut hadits shahih yang diriwayatkan Imam Bukhari, Rasulullah Muhammad SAW mengerjakan sholat berkiblat ke Al-Quds selama sekitar 16 atau 17 bulan semasa berada di Madinah. Dalam sejarah Islam, arah kiblat memang pernah diubah. Setelah semula mengarah ke Masjidil Aqsha (Al-Quds), kemudian turun firman ALLAH SWT untuk mengubah arah kiblat ke arah ka'bah seperti diabadikan dalam Al-Qur'an surat Al-Baqarah ayat 144:

“Sungguh Kami (sering) melihat wajahmu menengadah ke langit, maka sungguh Kami akan palingkan kamu ke kiblat yang kamu sukai. Maka palingkanlah wajahmu kearah Masjidil Haram”. (Muhammad Abdul Hamid Asy-Syarqawi & Muhammad Raja'I Ath-Thahlawi : 2009)

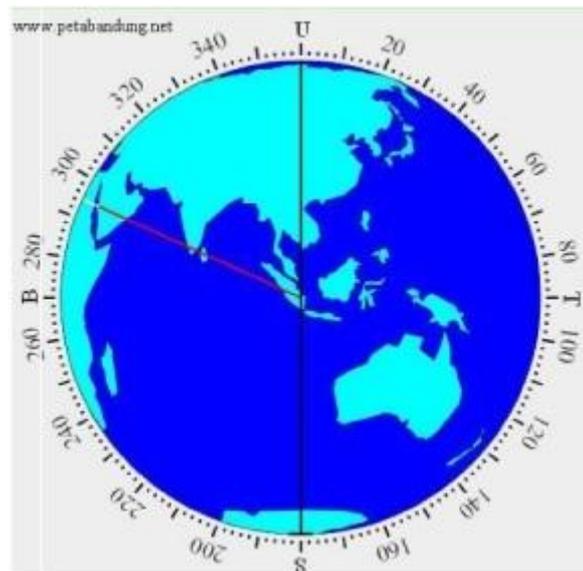
2.1.1 Menentukan Arah Kiblat

Arah Kiblat dapat dengan mudah ditentukan apabila Ka'bah dapat terlihat langsung oleh kita. Putarlah tubuh sehingga pandangan lurus kita menghadap ke Ka'bah. Ini dapat kita lakukan apabila kita berada di Mekah, kota di mana Ka'bah berada. Dalam 1000 tahun terakhir, sejumlah matematikawan dan astronom muslim seperti Biruni telah melakukan perhitungan yang tepat untuk menentukan arah kiblat dari berbagai tempat di dunia. Seluruhnya setuju bahwa setiap tahun ada dua hari dimana matahari berada tepat di atas ka'bah, dan arah bayangan matahari dimanapun di dunia pasti mengarah ke kiblat. Peristiwa tersebut terjadi

setiap tanggal 28 Mei pukul 9.18 GMT (16.18 WIB) dan 16 Juli jam 9.27 GMT (16.27 WIB) untuk tahun biasa. Sedang kalau tahun kabisat, tanggal tersebut dimajukan satu hari, dengan jam yang sama. Tentu saja pada waktu tersebut hanya separuh dari bumi yang mendapat sinar matahari. Selain itu terdapat 2 hari lain dimana matahari tepat di "balik" ka'bah (antipoda), dimana bayangan matahari pada waktu tersebut juga mengarah ke ka'bah. Peristiwa tersebut terjadi pada tanggal 28 November 21.09 GMT (4.09 WIB) dan 16 Januari jam 21.29 GMT (4.29 WIB). (Wikipedia : 2016)

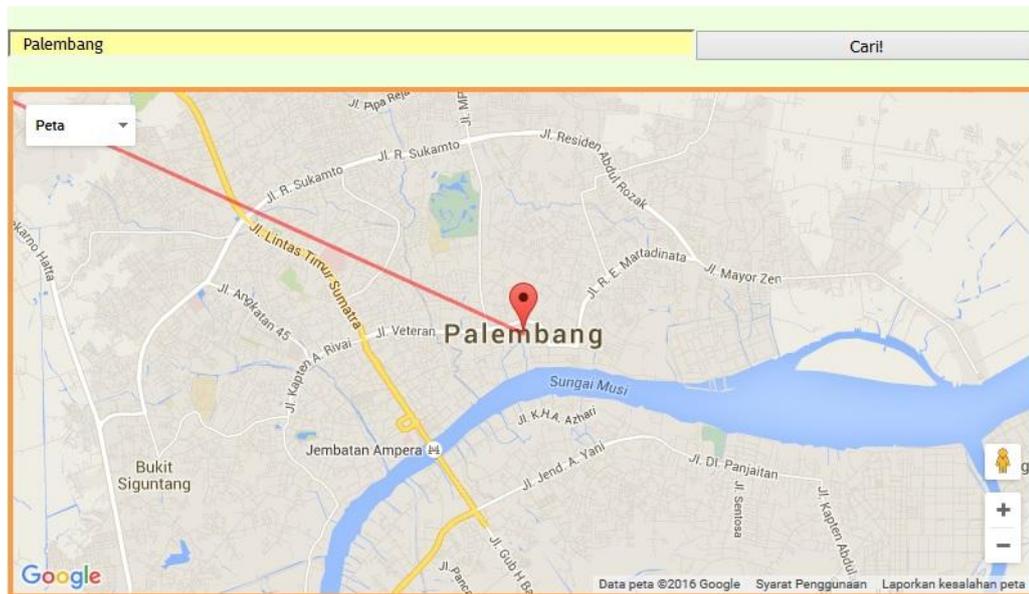


Gambar 2.1 Perhitungan Geometris Arah Kiblat
(Sumber : Wikipedia.com)



Gambar 2.2 Arah Kiblat di Palembang, Sumatra Selatan
(Sumber : petabandung.net/kiblat/)

- Referensi Panduan Arah Kiblat Berdasarkan Website



Gambar 2.3 Posisi Arah Kiblat Berdasarkan Kalibrasi Google Map

(Sumber : al-habib.info/arah-kiblat/)

Gambar 2.4 Konfigurasi Kalibrasi Arah dan Jarak Kiblat Posisi Palembang, Sumatra Selatan

(Sumber : al-habib.info/arah-kiblat/)



Sebuah layanan, mirip dengan *Qibla Locator*, yang memanfaatkan Peta Google untuk mengetahui dengan mudah arah kiblat dari semua titik atau tempat di muka bumi. Nama kota atau tempat dapat dicari dan dengan satu klik pada peta, arah kiblat akan digambarkan dengan tambahan informasi tentang berapa derajat arah sebenarnya dari utara.

2.2 Arduino

Arduino adalah pengendali *mikro single-board* yang bersifat *open-source*, diturunkan dari *Wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardware*nya memiliki prosesor Atmel AVR dan *software*nya memiliki bahasa pemrograman sendiri. (Wikipedia : 2016)

Arduino juga merupakan *platform hardware* terbuka yang ditujukan kepada siapa saja yang ingin membuat purwarupa peralatan elektronik interaktif berdasarkan *hardware* dan *software* yang fleksibel dan mudah digunakan. Mikrokontroler diprogram menggunakan bahasa pemrograman arduino yang memiliki kemiripan syntax yang diterapkan pada bahasa pemrograman C. Karena sifatnya yang terbuka maka siapa saja dapat mengunduh skema *hardware* arduino dan membangunnya.

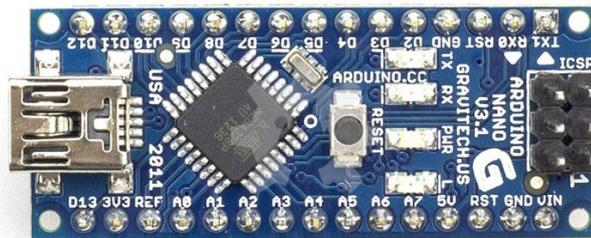
Arduino menggunakan keluarga mikrokontroler ATmega yang dirilis oleh Atmel sebagai basis, namun ada individu/perusahaan yang membuat *clone* arduino dengan menggunakan mikrokontroler lain dan tetap kompatibel dengan arduino pada level *hardware*. Untuk fleksibilitas, program dimasukkan melalui *bootloader* meskipun ada opsi untuk *bypass bootloader* dan menggunakan *downloader* untuk memprogram mikrokontroler secara langsung melalui *port ISP*.

2.3 Arduino Nano

Arduino Nano adalah salah satu papan pengembangan mikrokontroler yang berukuran kecil, lengkap dan mendukung penggunaan *breadboard*. Sepertinya namanya, Nano yang berukuran kecil dan sangat sederhana ini, menyimpan banyak fasilitas. Sudah dilengkapi dengan FTDI untuk pemrograman

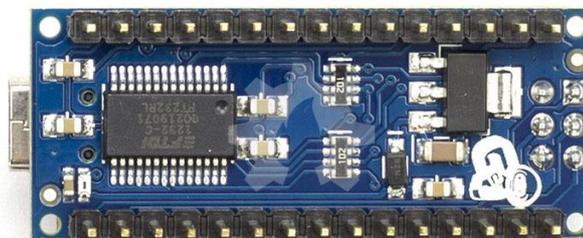
lewat *Micro* USB. 14 Pin I/O Digital, dan 8 Pin input Analog (lebih banyak dari Uno). (Wikipedia : 2016)

Arduino Nano diciptakan dengan basis mikrokontroler ATmega328 (untuk Arduino Nano versi 3.x) atau ATmega 168 (untuk Arduino versi 2.x). Arduino Nano kurang lebih memiliki fungsi yang sama dengan Arduino Duemilanove, tetapi dalam paket yang berbeda. Arduino Nano tidak menyertakan colokan DC berjenis *Barrel Jack*, dan dihubungkan ke komputer menggunakan *port* USB Mini-B. Arduino Nano dirancang dan diproduksi oleh perusahaan Gravitech. Dan ada yang menggunakan ATMEGA168, atau ATMEGA328, dengan *clock* 16MHz. (hendriono.com/blog/post/mengenal-arduino-nano : 2016)



Gambar 2.5 Arduino Nano Depan

(Sumber : hendriono.com/blog/post/mengenal-arduino-nano)



Gambar 2.6 Arduino Nano Belakang

(Sumber : hendriono.com/blog/post/mengenal-arduino-nano)



2.3.1 Spesifikasi Arduino Nano

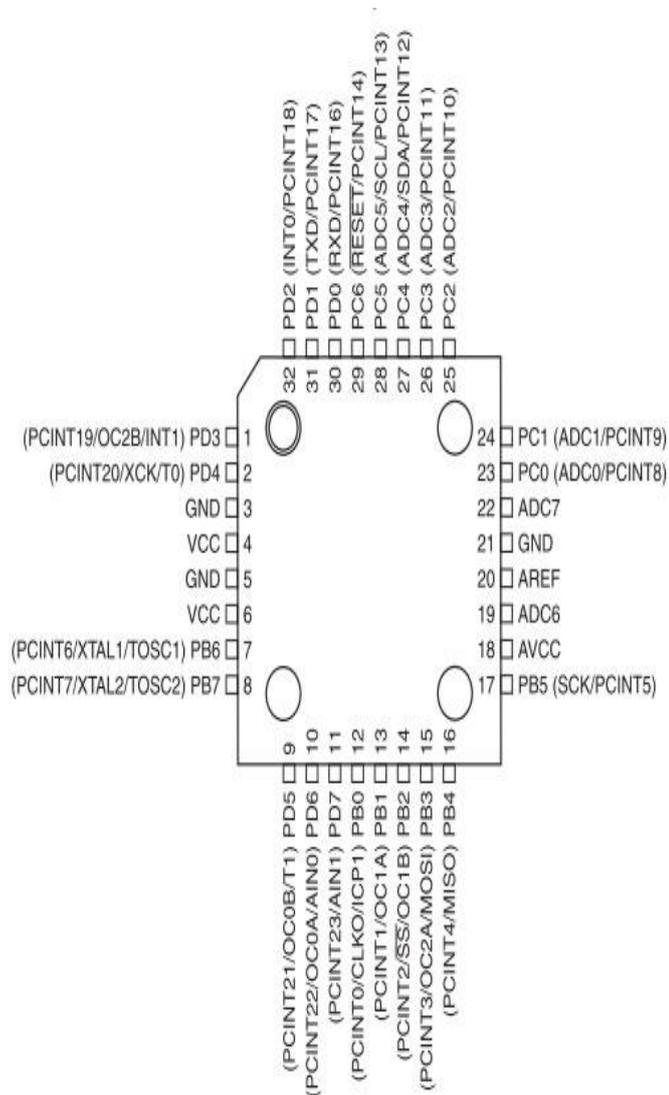
Dibawah ini spesifikasi dari Arduino Nano:

Tabel 2.1 Spesifikasi Arduino Nano

Mikrokontroler	Atmel ATmega168 atau ATmega328
Tegangan Operasi	5V
Input Voltage (disarankan)	7-12V
Input Voltage (limit)	6-20V
Pin Digital I/O	14 (6 pin digunakan sebagai output PWM)
Pins Input Analog	8
Arus DC per pin I/O	40 mA
Flash Memory	16KB (ATmega168) atau 32KB (ATmega328) 2KB digunakan oleh Bootloader
SRAM	1 KB (ATmega168) atau 2 KB (ATmega328)
EEPROM	512 byte (ATmega168) atau 1KB (ATmega328)
Clock Speed	16 MHz
Ukuran	1.85cm x 4.3cm

2.3.2 Pemetaan Pin

Dibawah ini pemetaan pin ATmega328 pada Arduino Nano :



Gambar 2.7 Pemetaan Pin ATmega328 SMD

(Sumber : hendriyono.com/blog/post/mengenal-arduino-nano)

Perhatikan pemetaan antara pin Arduino Nano dan port ATmega328 SMD. Pemetaan untuk ATmega8, ATmega168, dan ATmega328 sangat identik atau sama persis.

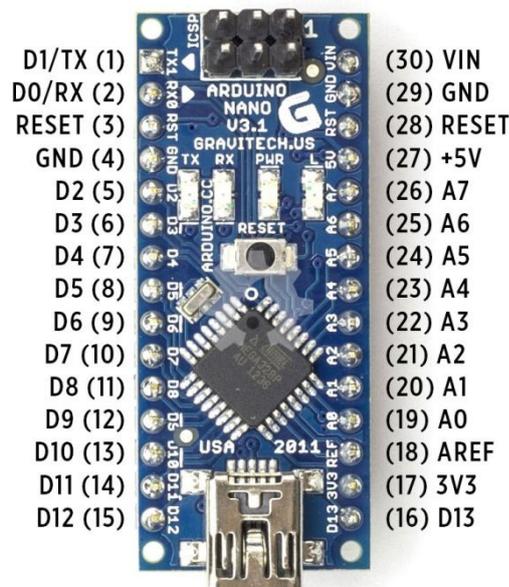


Tabel 2.2 Perbandingan Pemetaan Pin Arduino Nano dengan Mikrokontroler (ATmega328)

(Sumber : hendriono.com/blog/post/mengenal-arduino-nano)

Nomor Pin	Nama Pin	Nomor Pin	Nama Pin
ATmega328		Arduino Nano	
1	PD3 (PCINT19/OCB2B/INT1)	6	Digital Pin 3 (PWM)
2	PD4 (PCINT20/XCK/T0)	7	Digital Pin 4
3	GND	4 & 29	GND
4	VCC	27	VCC
5	GND	4 & 29	GND
6	VCC	27	VCC
7	PB6 (PCINT6/XTAL1/TOASC1)	-	-
8	PB7 (PCINT7/XTAL2/TOASC2)	-	-
9	PD5 (PCINT21/OC0B/T1)	8	Digital Pin 5 (PWM)
10	PD6 (PCINT22/OC0A/AIN0)	9	Digital Pin 6 (PWM)
11	PD7 (PCINT23/AIN1)	10	Digital Pin 7
12	PB0 (PCINT0/CLK0/ICP1)	11	Digital Pin 8
13	PB1 (PCINT1/OC1A)	13	Digital Pin 9 (PWM)
14	PB2 (PCINT2/SS/OC1B)	13	Digital Pin 10 (PWM - SS)
15	PB3 (PCINT3/OC2A/MOSI)	14	Digital Pin 11 (PWM - MOSI)
16	PB4 (PCINT4/MISO)	15	Digital Pin 12 (MISO)
17	PB5 (PCINT5/SCK)	16	Digital Pin 13 (SCK)

18	AVCC	27	VCC
19	ADC6	25	Analog Input 6
20	AREF	18	AREF
21	GND	4 & 29	GND
22	ADC7	26	Analog Input 7
23	PC0 (PCINT8/ADC0)	19	Analog Input 0
24	PC1 (PCINT9/ADC1)	20	Analog Input 1
25	PC2 (PCINT10/ADC2)	21	Analog Input 2
26	PC3 (PCINT11/ADC3)	22	Analog Input 3
27	PC4 (PCINT12/ADC4/SDA)	24	Analog Input 4 (SDA)
28	PC5 (PCINT13/ADC5/SCL)	25	Analog Input 5 (SCL)
29	PC6 (PCINT14/RESET)	28 & 3	RESET
30	PD0 (PCINT16/RXD)	2	Digital Pin 0 (RX)
31	PD1 (PCINT17/TXD)	1	Digital Pin 1 (TX)
32	PD2 (PCINT18/INT0)	5	Digital Pin 2



Gambar 2.8 Pin Diagram Arduino Nano

(Sumber : hendriono.com/blog/post/mengenal-arduino-nano)



2.3.3 Pemrograman Arduino Nano

Pemrograman board Arduino Nano dilakukan dengan menggunakan Arduino *Software* (IDE) yang bisa di *download* gratis. *Chip* ATmega328 yang terdapat pada Arduino Nano telah diisi program awal yang sering disebut *bootloader*. *Bootloader* tersebut yang bertugas untuk memudahkan anda melakukan pemrograman lebih sederhana menggunakan Arduino *Software*, tanpa harus menggunakan tambahan *hardware* lain. Cukup hubungkan Arduino dengan kabel USB ke PC, Mac, atau Linux anda, jalankan Arduino *Software* (IDE), dan anda sudah bisa mulai memrogram *chip* ATmega328. Lebih mudah lagi, di dalam Arduino *Software* sudah diberikan banyak contoh program yang memanjakan dalam belajar mikrokontroller.

Untuk pengguna mikrokontroller yang sudah lebih mahir, dapat tidak menggunakan *bootloader* dan melakukan pemrograman langsung via *header* ICSP (*In Circuit Serial Programming*) dengan menggunakan Arduino ISP.

2.3.4 Power Supply Arduino Nano

Development Board Arduino Nano dapat diberi tenaga dengan *power* yang diperoleh dari koneksi kabel Mini-B USB, atau via *power supply eksternal*. *External power supply* dapat dihubungkan langsung ke pin 30 atau *Vin(unregulated 6V - 20V)*, atau ke pin 27 (*regulated 5V*). Sumber tenaga akan otomatis dipilih mana yang lebih tinggi tegangan

Beberapa pin *power* pada Arduino Nano :

- **GND**. Ini adalah *ground* atau negatif.
- **Vin**. Ini adalah pin yang digunakan jika anda ingin memberikan *power* langsung ke board Arduino dengan rentang tegangan yang disarankan 7V - 12V
- **Pin 5V**. Ini adalah pin *output* dimana pada pin tersebut mengalir tegangan 5V yang telah melalui regulator
- **3V3**. Ini adalah pin *output* dimana pada pin tersebut disediakan tegangan 3.3V yang telah melalui regulator



- **REF.** Ini adalah pin yang menyediakan referensi tegangan mikrokontroller. Biasanya digunakan pada *board shield* untuk memperoleh tegangan yang sesuai, apakah 5V atau 3.3V

2.3.5 Memori Arduino Nano

Chip ATmega328 pada Arduino Nano memiliki memori 32 KB, dengan 0.5 KB dari memori tersebut telah digunakan untuk *bootloader*. Jumlah SRAM 2 KB, dan EEPROM 1 KB, yang dapat di baca-tulis dengan menggunakan EEPROM *library* saat melakukan pemrograman.

2.3.6 Input dan Output (I/O)

Arduino Nano memiliki 14 buah digital pin yang dapat digunakan sebagai *input* atau *output*. Pin ini akan mengeluarkan tegangan 5V untuk mode *high* (logika 1) dan 0V untuk mode *low* (logika 0) jika dikonfigurasi sebagai pin *output*. Jika di konfigurasi sebagai pin *input*, maka ke 14 pin ini dapat menerima tegangan 5V untuk mode *high* (logika1) dan 0V untuk mode *low* (logika0).

Dengan menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digital(Read)*. Pin-pin tersebut bekerja pada tegangan 5V, dan setiap pin dapat menyediakan atau menerima arus 20mA, dan memiliki tahanan *pull-up* sekitar 20-50k ohm (secara *default* dalam posisi *disconnect*). Nilai maximum adalah 40mA, yang sebisa mungkin dihindari untuk menghindari kerusakan *chip* mikrokontroller

Beberapa pin memiliki fungsi khusus :

- **Serial**, terdiri dari 2 pin : pin 0 (RX) dan pin 1 (TX) yang digunakan untuk menerima (RX) dan mengirim (TX) data serial. Kedua pin ini terhubung langsung ke pin IC FTDI USB-TTL.
- **External Interrupts**, yaitu pin 2 dan pin 3. Kedua pin tersebut dapat digunakan untuk mengaktifkan *interrupts* dari sumber *eksternal*. Interupsi dapat terjadi ketika timbul kenaikan atau penurunan tegangan pada pin D₂ atau pin D₃. Gunakan fungsi *attachInterrupt()*



- **PWM:** Pin 3, 5, 6, 9, 10, dan 11 menyediakan *output* PWM (*Pulse Width Modulator*) 8-bit dengan menggunakan fungsi *analogWrite()*
- **SPI :** Pin 10 (SS), 11 (MOSI), 12 (MISO), dan 13 (SCK) mendukung komunikasi SPI dengan menggunakan *SPI Library*
- **LED :** Pin 13. Pada pin 13 terhubung *built-in* led yang dikendalikan oleh digital pin no 13.

Arduino Nano memiliki 8 buah *input* analog, yang diberi tanda dengan A0 hingga A7. Pin analog ini terhubung ke ADC (*analog to digital converter*) internal yang terdapat di dalam mikrokontroler. Masing-masing pin analog tersebut memiliki resolusi 1024 bits (jadi bisa memiliki 1024 nilai). Secara *default*, pin-pin tersebut diukur dari *ground* ke 5V pada arus searah dengan besar arus maksimum 40 mA, namun bisa juga menggunakan pin REF dengan menggunakan fungsi *analogReference()* lebar *range* ini dapat diubah dengan memberikan sebuah tegangan referensi dari luar melalui pin V_{ref} .

Pin analog selain dapat digunakan untuk input data analog, juga dapat digunakan sebagai pin digital I/O, kecuali pin analog A6 dan A7 tidak bisa dijadikan sebagai pin digital, hanya sebagai analog. Beberapa pin lainnya pada *board* ini adalah :

- **I2C :** Pin A4 (SDA) dan A5 (SCL). Pin ini mendukung komunikasi I2C (TWI) dengan menggunakan *Wire Library*.
- **AREF.** Sebagai referensi tegangan untuk *input* analog.
- **Reset.** Hubungkan ke *LOW* untuk melakukan *reset* terhadap mikrokontroler. Biasanya digunakan untuk dihubungkan dengan *switch* yang dijadikan tombol *reset*.

2.3.7 Komunikasi Arduino Nano

Arduino Nano 3.0 memiliki beberapa fasilitas untuk berkomunikasi dengan komputer, berkomunikasi dengan Arduino lainnya, atau dengan mikrokontroler lain nya. *Chip* Atmega328 menyediakan komunikasi serial UART TTL (5V) yang tersedia di pin 0 (RX) dan pin 1 (TX). Sebuah *chip* FTDI yang



terdapat pada *board* yang dapat dihubungkan langsung ke komputer berfungsi menterjemahkan bentuk komunikasi ini melalui USB dan akan tampil sebagai *Virtual com-Port* pada *operating* sistem di komputer.

Pada *Arduino Software (IDE)* terdapat monitor serial yang memudahkan programmer untuk menampilkan data serial sederhana (data tekstual) untuk dikirim / diterima menuju *Arduino* atau keluar dari *Arduino*. Lampu led TX dan RX akan menyala berkedip-kedip ketika ada data yang ditransmisikan (terjadi komunikasi data) melalui *chip* FTDI USB to Serial via kabel USB ke komputer. Untuk menggunakan komunikasi serial dari digital pin, gunakan *SoftwareSerial library*.

Selain dapat berkomunikasi dengan menggunakan data serial melalui *virtual com-port*, *Arduino Nano* juga dilengkapi dengan mode komunikasi I2C (TWI) dan SPI untuk komunikasi antar *hardware*. *Chip* ATmega328 juga mendukung komunikasi I2C (TWI) dan SPI. Di dalam *Arduino Software (IDE)* sudah termasuk *Wire Library* untuk memudahkan anda menggunakan bus I2C. Untuk menggunakan komunikasi SPI, gunakan *SPI library*.

2.3.8 *Reset Otomatis (software)*

Biasanya, ketika melakukan pemrograman mikrokontroler, harus menekan tombol *reset* sesaat sebelum melakukan *upload* program. Pada *Arduino Nano*, hal ini tidak lagi merepotkan. *Arduino Nano* telah dilengkapi dengan *auto reset* yang dikendalikan oleh *software* pada komputer yang terkoneksi. Salah satu jalur *flow control* (DTR) dari ATmega16U pada *Arduino Nano R3* terhubung dengan jalur *reset* pada ATmega328 melalui sebuah kapasitor 100nF. Ketika jalur tersebut diberi nilai *low*, mikrokontroler akan di *reset*. Dengan demikian proses *upload* akan jauh lebih mudah dan tidak harus menekan tombol *reset* pada saat yang tepat seperti biasanya.



2.4 Kompas

Kompas adalah alat navigasi untuk menentukan arah berupa sebuah panah penunjuk magnetis yang bebas menyelaraskan dirinya dengan medan magnet bumi secara akurat. Kompas memberikan rujukan arah tertentu, sehingga sangat membantu dalam bidang navigasi. Arah mata angin yang ditunjuknya adalah utara, selatan, timur, dan barat. Apabila digunakan bersama-sama dengan jam dan sekstan, maka kompas akan lebih akurat dalam menunjukkan arah. Alat ini membantu perkembangan perdagangan maritim dengan membuat perjalanan jauh lebih aman dan efisien dibandingkan saat manusia masih berpedoman pada kedudukan bintang untuk menentukan arah.

Penemuan bahwa jarum magnetik selalu mengarah ke utara dan selatan terjadi di Cina dan diuraikan dalam buku *Loven Heng*. Pada abad kesembilan, orang Cina telah mengembangkan kompas berupa jarum yang mengambang dan jarum yang berputar. Pelaut Persia memperoleh kompas dari orang Cina dan kemudian memperdagangkannya. Tetapi baru pada tahun 1877 orang Inggris, William Thomson, 1st Baron Kelvin (Lord Kelvin) membuat kompas yang dapat diterima oleh semua negara. Dengan memperbaiki kesalahan-kesalahan yang timbul dari deviasi magnetik karena meningkatnya penggunaan besi dalam arsitektur kapal.

Alat apa pun yang memiliki batang atau jarum magnetis yang bebas bergerak menunjuk arah utara magnetis dari magnetosfer sebuah planet sudah bisa dianggap sebagai kompas. Kompas jam adalah kompas yang dilengkapi dengan jam matahari. Kompas variasi adalah alat khusus berstruktur rapuh yang digunakan dengan cara mengamati variasi pergerakan jarum. Girokompas digunakan untuk menentukan utara sejati.

Lokasi magnet di Kutub Utara selalu bergeser dari masa ke masa. Penelitian terakhir yang dilakukan oleh *The Geological Survey of Canada* melaporkan bahwa posisi magnet ini bergerak kira-kira 40 km per tahun ke arah barat laut.

▪ Kompas analog

Kompas analog adalah kompas yang biasa kita lihat dalam kehidupan sehari-hari. Misalnya saja kompas yang dipakai ketika acara pramuka. Sedangkan kompas digital merupakan kompas yang telah menggunakan proses digitalisasi. Dengan kata lain cara kerja kompas ini menggunakan komputerisasi.

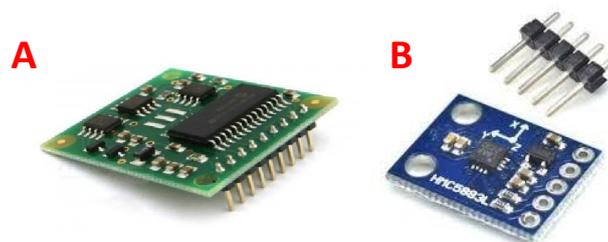


Gambar 2.9 Kompas Analog (Manual)

▪ Kompas digital

Diciptakannya kompas digital bertujuan untuk melengkapi kebutuhan robotika yang semakin canggih. Dunia robotika ini sangat membutuhkan alat navigasi yang efektif dan efisien. Sementara itu alat sistem navigasi yang tersedia di pasaran harganya mahal. Sedangkan kompas sendiri merupakan sebuah alat sistem navigasi yang efektif dengan harga lebih murah. Oleh karena itu kompas digital diharapkan bisa mensubstitusi alat sistem navigasi pada robot. Selain di bidang robotika pengembangan teknologi elektronika dengan memanfaatkan kompas digital juga dapat diterapkan. Kompas-kompas digital yang ada di pasaran banyak macamnya. Di antaranya yaitu CMPS03 dan HMC5883L, dll.

Ada dua cara untuk memperoleh informasi arah dari kompas digital ini yaitu dengan membaca sinyal PWM (Pulse Width Modulation) pada pin 4 atau dengan membaca data *interface* I2C pada pin 2 dan 3. Sinyal PWM adalah sebuah sinyal yang telah dimodulasi lebar pulsanya.



Gambar 2.10 Kompas Digital (A). CMPS03, (B). HMC5883L



2.5 Sensor Kompas HMC5883L

HMC5883L adalah sebuah sensor yang digunakan untuk menunjukkan arah mata angin, atau bisa juga disebut sebagai kompas digital. Sensor ini menggunakan komponen utama berupa IC HMC5883L yang merupakan IC kompas digital 3 axis yang memiliki *interface* berupa 2 pin I2C. HMC5883L memiliki sensor *magneto-resistive* HMC118X *series* ber-resolusi tinggi, ditambah ASIC dengan konten *amplification*, *automatic degaussing strap driver*, *offset cancellation* dan 12 bit ADC yang memungkinkan keakuratan kompas mencapai 1 sampai 2 derajat. Modul ini biasa digunakan untuk keperluan sistem navigasi otomatis, *mobile phone*, *netbook* dan perangkat navigasi personal. Modul ini memiliki 5 pin, diantaranya adalah VCC, Gnd, SDA, SCL, dan DRDY.

HMC5883L menggunakan tiga unsur *magneto-resistif*. Salah satunya adalah dapat mengubah perlawanan secara proporsional dengan kekuatan medan magnet di sepanjang sumbunya. Poin penting yang perlu dicatat bahwa kepekaan setiap elemen individu *magneto-resistif* merupakan komponen yang sejajar antara medan magnet dengan sumbu elemen. Ketiga unsur dalam paket sensor yang berorientasi sedemikian rupa sehingga masing-masing adalah *orthogonal* dengan dua lainnya. Dengan kata lain, masing-masing sumbu X, Y dan Z adalah searah. Dengan demikian, ketika paket sensor terkena medan magnet, kekuatan dan arah medan yang dalam ruang tiga-dimensi dapat ditentukan dari resistensi yang ditunjukkan oleh tiga elemen. HMC5883L menyajikan kepekaan medan magnet sebagai vektor terhadap ketiga sumbu yang ditandai dengan 16-bit (satu untuk setiap sumbu). Hal ini juga secara otomatis mengkompensasi setiap ketergantungan internal terhadap pengaruh *offset* dan sudut pada masing-masing elemen.

HMC5883L adalah sensor magnet terkemas dalam *surface mount* 3.0x3.0x0.9 mm 16-pin *leadless chip carrier* (LCC). Menggunakan teknologi anisotropic *magneto-resistive* (AMR) *Honeywell*, HMC5883L menyediakan kepresisian lebih pada sensitifitas dan linieritas sumbu dan dirancang untuk mengukur kedua arah dan medan magnet bumi. *Port* I2C ini adalah *port* I/O dari



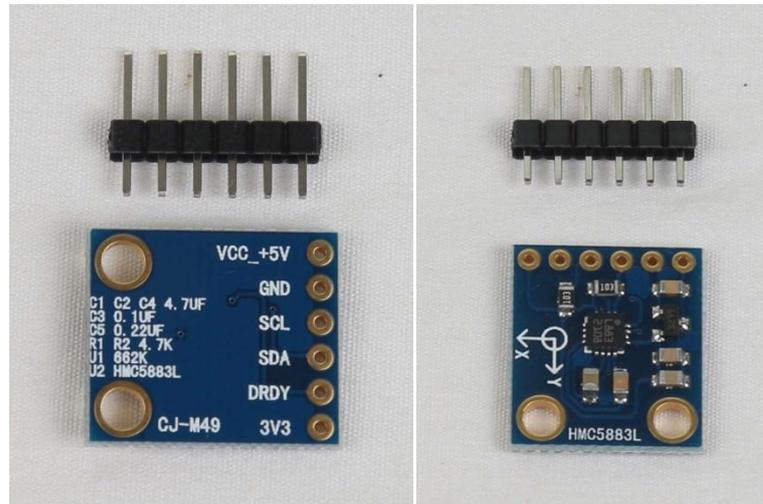
sensor ini, jika dihubungkan ke port mikrokontroler maka dihubungkan ke *PortC.0* (SCL) dan *PortC.1* (SDA).

Spesifikasi:

- ADC 12-bit ADC terkopling dengan sensor AMR *low noise* yang akan menghasilkan resolusi 2-milli gauss pada medan ± 8 gauss
- Mengijinkan akurasi kompas 1 sampai 2 derajat
- Tersedia *self-test* yaitu fitur tambahan yang dikemas dalam ASIC yang dapat digunakan untuk:
 - Secara cepat menguji fungsi-fungsi sensor tanpa perlu membutuhkan peralatan pengujian yang mahal.
 - Pencocokan sensitifitas dari sumbu/sensor yang berbeda
 - Menyetel pergeseran sensitifitas karena suhu
- Tegangan kerjanya rendah (2.16 ~ 3.6V) dan konsumsi daya rendah (100 uA). Cocok untuk aplikasi yang dicatu menggunakan *battery*
- Tersedia rangkaian *drive strap*
- Menyediakan demagnetisasi sensor untuk setiap pengukuran, dan juga kompensasi *offset* untuk mendapatkan pengukuran yang konsisten dengan akurasi hingga 1 sampai 2 derajat dan mereduksi perlunya kalibrasi ulang
- Antarmuka digital I2C
- *Range* medan magnet yang dapat diukur cukup lebar (± 8 Oe)
- Sensor bisa digunakan pada lingkungan dengan medan magnet yang kuat dengan akurasi kompas 1 sampai 2 derajat

Modul ini memiliki 5 pin, diantaranya :

1. VCC (5V)
2. GND
3. SCL
4. SDA
5. DRDY



Gambar 2.11 Sensor Kompas HMC5883L

(Sumber : bengkel-elektro.com/product.php?category=93&product_id=217)

Teori Kerja Sensor Kompas HMC5883L

Kompas merupakan salah satu alat penting dalam navigasi yang berfungsi sebagai penunjuk arah berdasarkan posisi kutub bumi. Sensor kompas HMC5883L sebuah kompas yang bekerja dengan menyelaraskan medan magnet bumi. Karena jarum kompas terbuat dari bahan besi, yang sejalan dengan ayunan pada bantalan di pusat seperti medan magnet bumi menariknya ke dalam keselarasan. Medan magnet ini berkembang di seluruh permukaan bumi sehingga dapat digunakan untuk membantu dalam menunjuk arah mata angin. *Magnetometer* menggunakan medan magnet tersebut, namun tidak menarik pada jarum kecil di dalamnya. Di dalam *magnetometer* terdapat tiga sensor magneto-resistif pada tiga sumbu. Hal tersebut menjelaskan bahwa efek medan magnet pada sensor ini mengatur aliran arus melalui sensor dengan menerapkan skala (milli-gauss). *The Honeywell HMC5883L* adalah permukaan *-mount*, modul multi-*chip* yang dirancang untuk medan magnet rendah penginderaan dengan antarmuka digital untuk aplikasi seperti *compassing* dan *magnetometry*. HMC5883L termasuk resolusi tinggi seri HMC118X *magneto-resistif* sensor dan ditambah amplifikasi ASIC. HMC5883L ini memanfaatkan *Anisotropic*



Magneto-resistive (AMR) teknologi *Honeywell* yang memberikan keuntungan lebih dari teknologi sensor magnetik lainnya. Ini *anisotropic, directional* sensor memiliki presisi dalam sumbu sensitivitas dan linearitas. Konstruksi *solid-state* sensor ini dengan sensitivitas *cross-* sumbu yang sangat rendah dirancang untuk mengukur baik arah dan besarnya medan magnet bumi, dari mili-gauss sampai 8 gauss. Sensor *Magnetic Honeywell* adalah salah satu sensor medan rendah paling sensitif dan dapat diandalkan dalam industri.

Sensor HMC5883L ini adalah sensor yang sangat sensitif sekali terhadap rotasi dan arah hadap sensor, dikarenakan sensor ini menggunakan medan magnet sebagai acuan dari pendeteksiannya / sebuah modul yang digunakan untuk menunjukkan arah mata angin digital, atau juga disebut kompas digital. Modul ini menggunakan komponen utama berupa IC HMC5883 yang merupakan IC kompas digital 3 *axis* yang memiliki *interface* berupa 2 pin I2C. Sensor yang memiliki respon terhadap rotasi atau putaran, jadi sensor ini akan memiliki nilai yang berbeda saat dia berada dengan posisi hadap yang berbeda, misal jika sensor ini menghadap ke utara dengan ke selatan, maka hasilnya saat posisi menghadap ke utara akan berbeda dengan pada saat sensor menghadap ke posisi selatan, begitulah teori kerja sensor kompas HMC5883L.

2.6 LCD Nokia 3310

Menampilkan konten pada layar alfanumerik normal adalah sangat terbatas, karena harus dibatasi dengan ukuran *font* dan tidak bisa menggambar grafik apapun juga, tapi konvensi grafik lcd benar-benar sangat mahal solusi dari semua itu dengan menggunakan Nokia 3310 monokrom LCD untuk menampilkan teks *font* yang besar dan grafis. alasan di balik menggunakan LCD ini. LCD ini juga dapat didukung dengan pasokan 3 volt sehingga benar-benar baik untuk baterai aplikasi bertenaga. Serta LCD ini juga mendukung untuk digunakan pada hampir semua mikrokontroler (dengan kemampuan untuk bekerja pada 3v) melakukan isi tampilan pada LCD ini. LCD Nokia 3310 ini memiliki resolusi 84x48 pixel dengan menggunakan PCD8544 *Chip* pengontrol Dari Philips. Ini adalah *chip-on kaca* (COG) dengan 8 pin konektor di sisi belakang LCD.



LCD (*Liquid Crystal Diode*) / *display* ini digunakan di Nokia 5110/3310 ponsel lama sebelum berubah ke mode *smart-phone* berubah setiap ponsel. Ini adalah 84x48 pixel layar LCD monokrom (hitam putih). Ukuran LCD ini relatif kecil dan ringkas, hanya sekitar 1,5 "diagonal, tapi sangat mudah dibaca dan dengan *backlight* (cahaya latar) putih (bisa di ubah). Tampilan ini terbuat dari 84x48 piksel individu, sehingga dapat digunakan untuk grafis, teks atau bitmap. Dengan harga yang terbilang relatif murah, mudah digunakan, dan hanya memerlukan beberapa digital I/O pin dan membutuhkan daya cukup rendah serta untuk mengaktifkan layar, yaitu hanya diperlukan 3 sampai 5 pin *output* digital (tergantung pada apakah ingin secara manual mengontrol *chip* pilih dan *reset* garis). pin lain dapat digunakan untuk mengendalikan (melalui *on / off* atau PWM) lampu latar LED *backlight* yang ditransfer ke transistor.

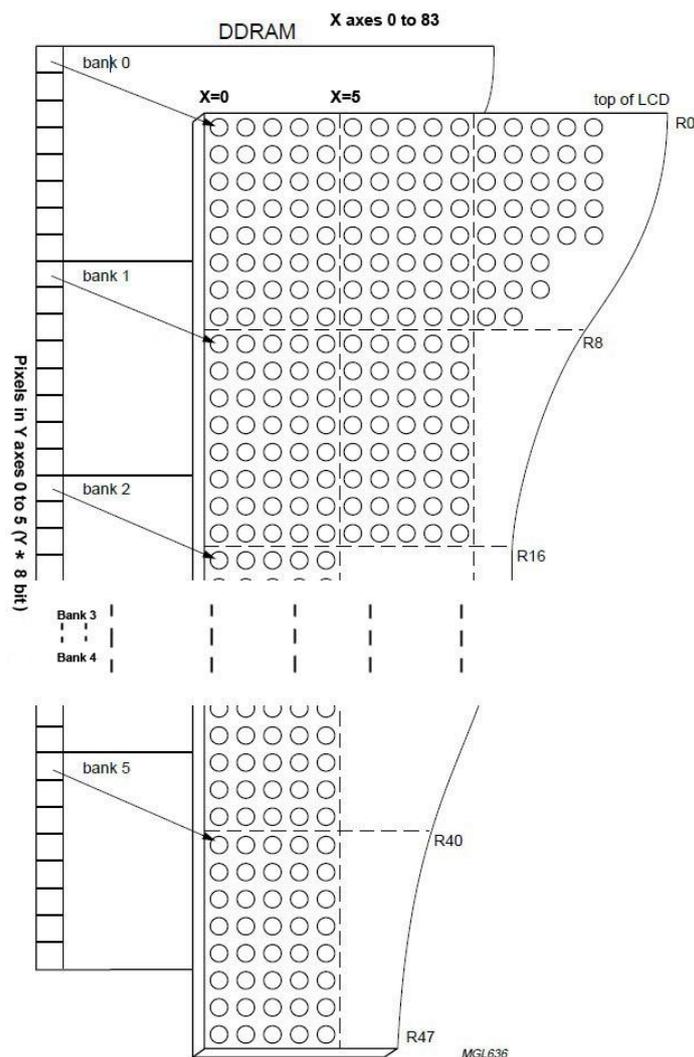
LCD Nokia3310 menggunakan kontroler PCD8544, yang sama digunakan dalam LCD Nokia 5110. PCD8544 adalah kekuatan CMOS LCD *controller / driver* rendah, yang dirancang untuk mengarahkan tampilan grafis dari 48 baris dan 84 kolom. Semua fungsi yang diperlukan untuk layar yang disediakan dalam satu *chip*, termasuk *on-chip* generasi pasokan LCD dan bias tegangan, sehingga minimal komponen eksternal dan konsumsi daya yang rendah. PCD8544 antarmuka untuk mikrokontroler melalui antarmuka serial bus. Nokia 5110/3310 adalah grafis layar LCD dasar untuk banyak aplikasi. Ini pada awalnya ditujukan untuk sebagai layar ponsel. Program perangkat lunak untuk proyek ini akan ditulis dalam C.

2.6.1 Spesifikasi LCD Nokia 3310

- LCD grafik (*monochrome*) hitam putih
- Termasuk Jenis LCD LPH 7779
- Berukuran kecil / ringkas dilengkapi dengan *backlight* putih (bisa di ubah)
- Tegangan Operasi 2,7V – 3,3V
- Arus Operasi <5mA (*backlight off*), <20mA (*backlight on*)
- LCD ini di dalamnya telah terdapat sebuah IC *driver/controller* LCD 84x48 *pixel* (84 kolom dan 48 baris) 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.
- LCD ini memiliki 84x48 pixel, yang dipetakan langsung dalam memori ram dari LCD (84x 48 bits), seperti yang ditunjukkan pada gambar dibawah:

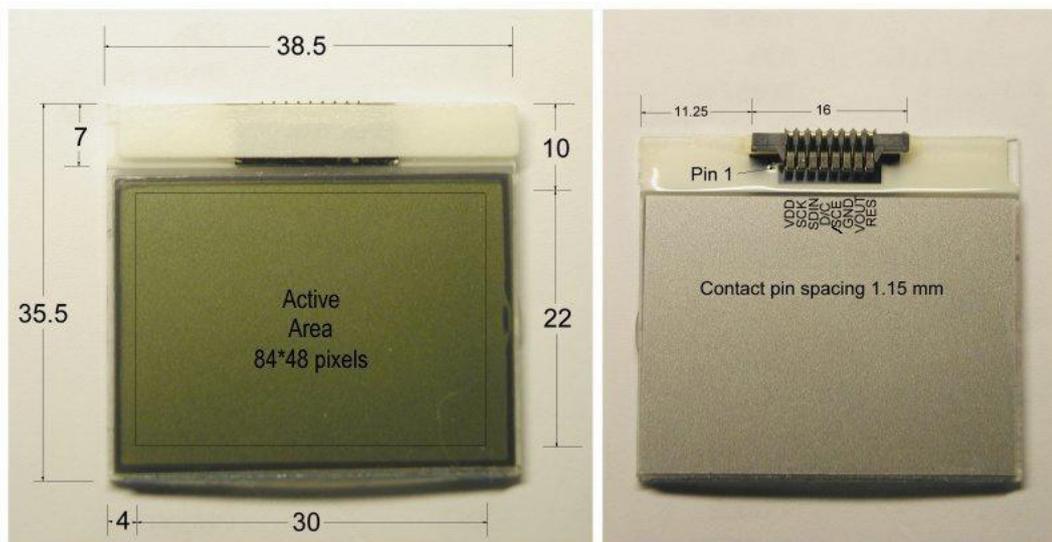


DDRAM to display mapping.

Gambar 2.12 Pemetaan 84x48 Pixel LCD Nokia 3310
(Sumber : circuitvalley.com/2011/08/nokia-3310-lcd-interfacing-with.html)

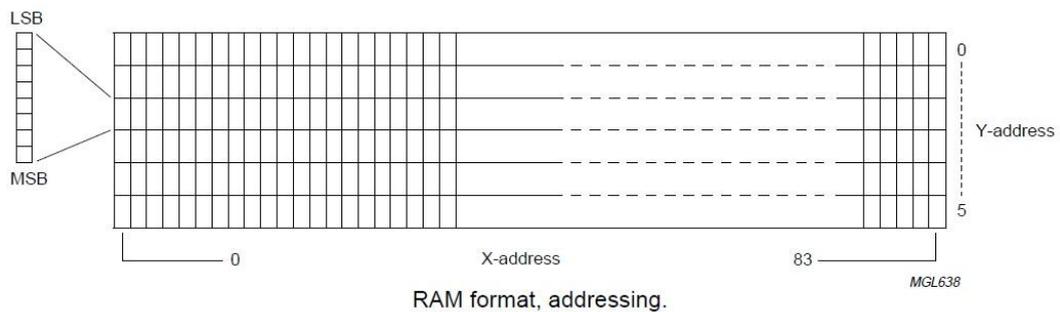


Gambar 2.13 LCD Nokia 3310
(Sumber : george-smasmart.co.uk/)



Gambar 2.14 Skala Ukuran dan Pin LCD Nokia 3310
(Sumber : circuitvalley.com/2011/08/nokia-3310-lcd-interfacing-with.html)

Contoh khas RAM ditunjukkan pada gambar dibawah, Daerah sumbu vertikal ditujukan 0-5 dengan delapan bit untuk setiap alamat ketika menggabungkan dengan sumbu x, dapat direpresentasikan sebagai bank.



Gambar 2.15 Penderskripsi Pengalamatan RAM LCD Nokia 3310

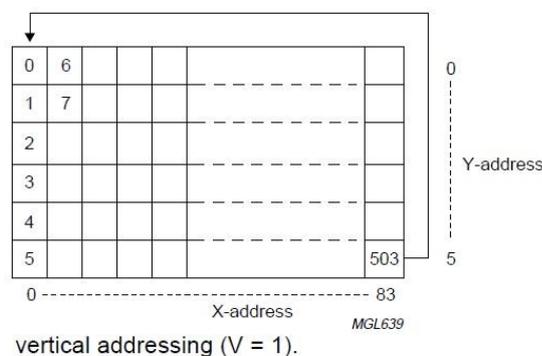
(Sumber : circuitvalley.com/2011/08/nokia-3310-lcd-interfacing-with.html)

- Sumbu horisontal ditujukan formulir 0-83 dan setiap bit akan merujuk sesuai *pixel* dalam arah X.
- Sebagai komponen eksternalnya perlu ditambahkan sebuah kapasitor dengan nilai minimal sebesar $1\mu\text{F}$ dari V_{out} ke *ground*.
- Pada LCD grafik Nokia 3310 ini tidak memiliki *on-chip character-set*.

2.6.2 Mode Pengalamatan (*Addressing Mode*)

Ada dua jenis mode pengalamatan di LCD ini

- Mode Pengalamatan Vertikal

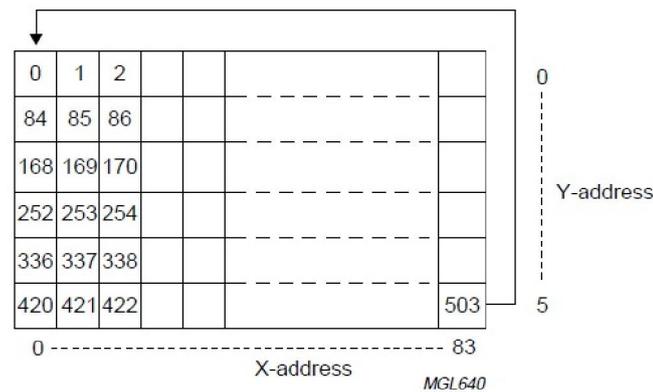


Gambar 2.16 Pengalamatan Mode Vertikal LCD Nokia 3310

(Sumber : circuitvalley.com/2011/08/nokia-3315-lcd-interfacing-with.html)

Mode pengalamatan vertikal setiap instruksi tulis akan kenaikan alamat arah Y dan kemudian kembali ke alamat awal

- Mode Pengalamatan Horizontal



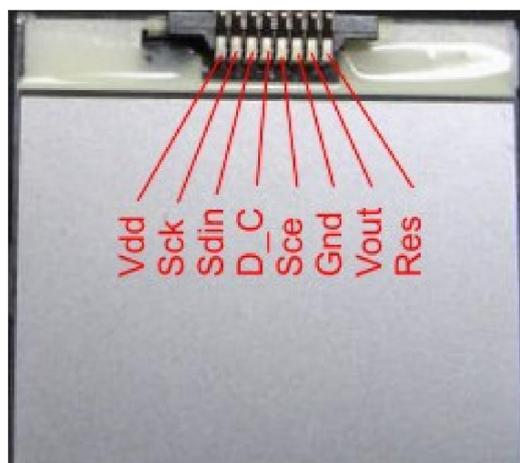
horizontal addressing ($V = 0$).

Gambar 2.17 Pengalamatan Mode Horizontal LCD Nokia 3310

(Sumber : circuitvalley.com/2011/08/nokia-3315-lcd-interfacing-with.html)

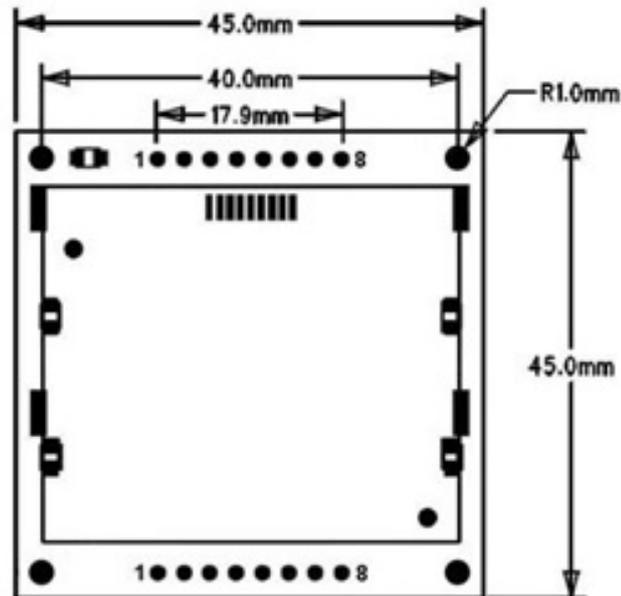
horizontal menangani setiap instruksi jalur akan kenaikan alamat arah X dan kembali ke alamat awal

2.6.3 Konfigurasi Pin LCD Nokia 3310



Gambar 2.18 Pendeskripsian Pin LCD Nokia 3310

(Sumber : iq-technologies.net/projects/mcu/013/NokiaLcd-konektor.jpg)



Gambar 2.19 Pin Diagram LCD Nokia 3310

(Sumber : shop.boxtec.ch/images/45040_a.jpg)

Kegunaan Pin LCD Nokia 3310

Pin 1 VCC /Vdd	: Pin untuk <i>supply</i> 2,7 sampai 3,3 Volt
Pin 2 SCLK	: Pin untuk <i>input</i> sinyal <i>clock</i>
Pin 3 SDIN	: Pin untuk <i>input</i> data serial
Pin 4 D/C	: Pin mode <i>select</i> (data atau <i>command</i>) (<i>LOW</i> menulis <i>command</i> (perintah) dan <i>HIGH</i> menulis data)
Pin 5 SCE	: Pin <i>input chip enable</i> aktif <i>LOW</i>
Pin 6 GND	: Pin untuk <i>ground</i>
Pin 7 Vout	: Pin <i>output</i>
Pin 8 RES	: Pin untuk <i>reset</i> aktif <i>LOW</i>
Pin LED	: Pin untuk <i>backlight</i> LCD aktif 2,7 sampai 3.2 Volt



2.6.4 Instruksi set PCD8544

Tabel 2.3 LCD Nokia 3310 *Interfacing* Mikrokontroler PCD8544 *Instruction set*

(Sumber : circuitvalley.com/2011/08/nokia-3315-lcd-interfacing-with.html)

Instruction set

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	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
	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 2.4 Penjelasan Simbol *Interfacing* LCD Nokia 3310 Mikrokontroler PCD8544 *Instruction Set*

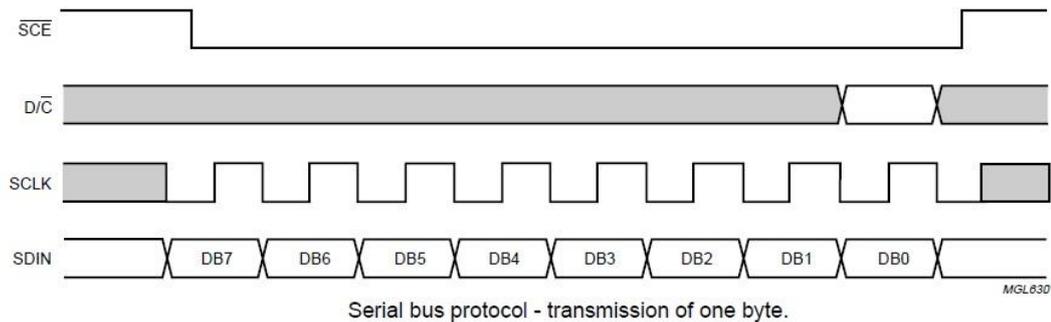
(Sumber : circuitvalley.com/2011/08/nokia-3315-lcd-interfacing-with.html)

Explanations of symbols in Table 1

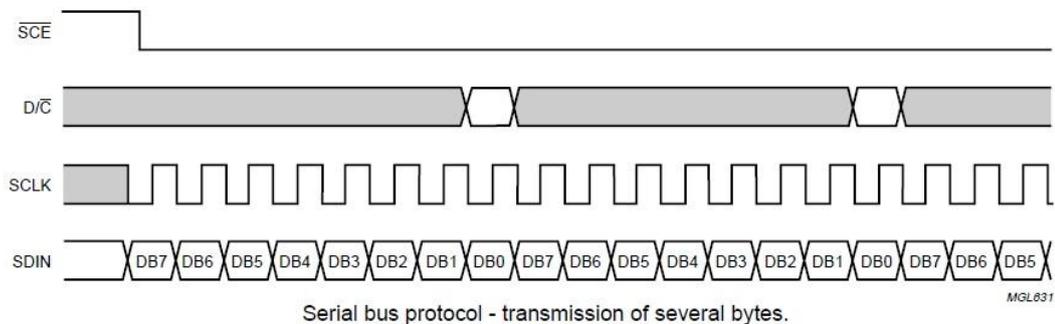
BIT	0	1
PD	chip is active	chip is in Power-down mode
V	horizontal addressing	vertical addressing
H	use basic instruction set	use extended instruction set
D and E	display blank normal mode all display segments on inverse video mode	
TC ₁ and TC ₀	V _{LCD} temperature coefficient 0 V _{LCD} temperature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

2.6.5 Komunikasi Serial

Antarmuka (*interfaceing*) pada dasarnya antarmuka perangkat serial (SPI). protokol serial bus untuk transmisi dari satu byte ditunjukkan pada gambar berikut:



Gambar 2.20 *Interfaceing* LCD Nokia 3310 Mikrokontroler PCD8544 SPI 1
(Sumber : circuitvalley.com/2011/08/nokia-3315-lcd-interfacing-with.html)



Gambar 2.21 *Interfaceing* LCD Nokia 3310 Mikrokontroler PCD8544 SPI
(Sumber : circuitvalley.com/2011/08/nokia-3315-lcd-interfacing-with.html)

Sebuah byte dikirim ke LCD sebagai berikut:

- Mengatur SCE Untuk GND.
- Set D / C ke keadaan diperlukan (Data atau perintah).
- Place sedikit di SDIN baris.
- Membuat tinggi ke rendah transisi *input* CLK.
- Repeat langkah 3 dan 4 untuk tujuh bit yang tersisa.



2.6.6 Inisialisasi Urutan LCD

- Tarik SCE baris ke *Ground* Aktifkan LCD The.
- Set D / C pin rendah untuk mengirimkan perintah ke LCD.
- Menulis 0x21 Byte ke LCD pada Bus serial. Berikut LCD beroperasi di Fungsi mengatur *Command* Untuk diperpanjang set instruksi.
- Menulis 0xC8 Byte ke LCD pada Bus serial. Ini mengatur tegangan operasi (Vop) dari LCD.
- Menulis 0x06 Byte ke LCD pada Bus serial. Ini mengatur *coefficient* suhu.
- Menulis 0x13 Byte ke LCD pada Bus serial. Ini mengatur sistem bias LCD.
- Menulis 0x20 Byte ke LCD pada Bus serial. Ini memungkinkan LCD untuk beroperasi dalam fungsi mengatur perintah dengan instruksi dasar
- Tuliskan Byte 0x0C untuk LCD di tee seri Bus. Ini mengatur tampilan dalam mode normal.

Lcd inisialisasi rutin (*routine*)

```
void lcdInit()
{
    SCE=0;
    DC=0;
    spi(0x21);
    spi(0xC8);
    spi(0x06);
    spi(0x13);
    spi(0x20);
    spi(0x0C);
}
```

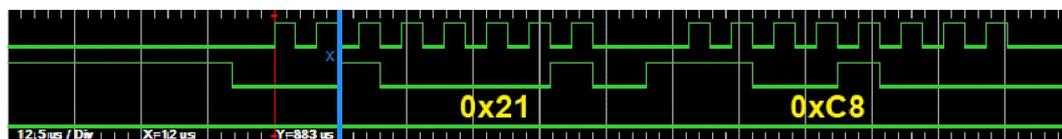
Clearing the LCD, setelah inisialisasi tarik SCE pin ke tanah untuk mengaktifkan

LCD.Set D / C pin ke tinggi untuk mengirim data ke LCD. Ulangi "tuliskan byte 0x00" operasi ke LCD untuk 504 kali (yaitu $54 \times 48 / 8$) ini akan mematikan semua piksel pada LCD.

Turn on all the pixels on the LCD, Tarik SCE pin ke tanah untuk mengaktifkan LCD. Set DC pin ke tinggi untuk mengirim data ke LCD. Ulangi "tuliskan byte 0xFF" operasi untuk LCD. Ulangi untuk LCD untuk 504 kali. Ini akan Aktifkan semua piksel pada LCD.

Display Character in LCD. tabel lookup 'FONTLOOKUP' untuk karakter ASCII termasuk dalam program. Tarik pin SCE ke tanah untuk mengaktifkan LCD. Set DC pin ke tinggi untuk mengirim data ke LCD menulis lima bytes dari 'FONTLOOKUP [X] [Y]' untuk LCD. Ini akan menghidupkan piksel untuk karakter ASCII pada LCD.

2.6.7 Pemanfaatan Logika LCD Nokia 3310



Gambar 2.22 Inisialisasi LCD Nokia 3310

(Sumber : circuitvalley.com/2011/08/nokia-3315-lcd-interfacing-with.html)

Fungsi berikut yang digunakan dalam program contoh

- Spi : *Routine* untuk jalur data ke LCD (menggunakan SPI modul di PIC 18F)
- lcdStr : *Routine* untuk menulis string ke LCD
- lcdDec : *Routine* untuk menulis angka ke LCD
- lcdDir : *Routine* untuk menulis langsung membentuk *table*
- gotoXy : *Routine* untuk mengatur X, poin Y di LCD
- delayMs : *Routine* untuk ms *delay*.

2.7 Baterai Ion – Lithium (Li-ion)

Baterai ion litium (biasa disebut Baterai Li-ion atau LIB) adalah salah satu anggota keluarga baterai isi ulang. Di dalam baterai ini, ion litium bergerak dari elektroda negatif ke elektroda positif saat dilepaskan, dan kembali saat diisi ulang. Baterai Li-ion memakai senyawa litium interkalasi sebagai bahan elektrodanya, berbeda dengan litium metalik yang dipakai di baterai litium non-isi ulang.

Baterai ion litium umumnya dijumpai pada barang-barang elektronik konsumen. Baterai ini merupakan jenis baterai isi ulang yang paling populer untuk peralatan elektronik portabel, karena memiliki salah satu kepadatan energi terbaik, tanpa efek memori, dan mengalami kehilangan isi yang lambat saat tidak digunakan. Selain digunakan pada peralatan elektronik konsumen, LIB juga sering digunakan oleh industri militer, kendaraan listrik, dan dirgantara. Seiring perjalanan waktu untuk memperbaiki teknologi LIB tradisional, berfokus pada kepadatan energi, daya tahan, biaya, dan keselamatan intrinsik.



Gambar 2.23 Baterai Ion – Lithium (Li-ion)

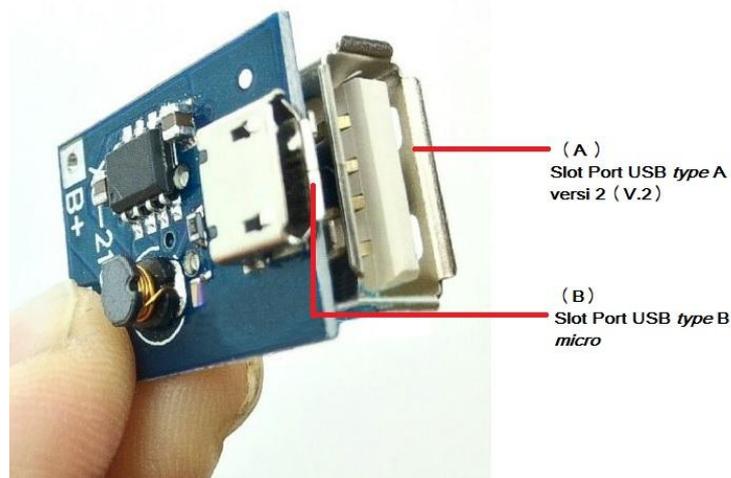
(Sumber : tokopedia.com)

Karakteristik kimiawi, kinerja, biaya, dan keselamatan jenis-jenis LIB cenderung bervariasi. Barang elektronik genggam biasanya memakai LIB berbasis litium kobalt oksida (LCO) yang memiliki kepadatan energi tinggi, namun juga memiliki bahaya keselamatan yang cukup terkenal, terutama ketika rusak. Litium besi fosfat (LFP), litium mangan oksida (LMO), dan litium nikel mangan kobalt oksida (NMC) memiliki kepadatan energi yang lebih rendah, tetapi hidup lebih lama dan keselamatannya lebih kuat. Bahan kimia ini banyak dipakai oleh

peralatan listrik, perlengkapan medis, dan lain-lain. NMC adalah pesaing utama di industri otomotif. Litium nikel kobalt alumunium oksida (NCA) dan litium titanat (LTO) adalah desain khusus yang ditujukan pada kegunaan-kegunaan tertentu.

2.8 Modul *Powerbank 1 Slot Multi Charger*

Sebuah modul berupa dual *slot port* USB, yang berupa *port* USB *type* A (versi 2) dan *port* USB *type* B (*micro*) dalam satu set kit modul. *Port* USB *type* A (v.2) merupakan *slot* jalur untuk mengisi daya (*charging*) alat ini sementara *port* yang satunya lagi yaitu *port* USB *type* B merupakan *slot* jalur *powerbank* untuk mengisi daya (*charging*) *handphone* pengguna.



Gambar 2.24 Modul *Powerbank 1 Slot Multi Charger*

(Sumber : tokopedia.com)

2.8.1 Fungsi Modul 1 *Slot Multi Charger*

- *Step Up*
- *Charger*
- Proteksi *Charger* 5v1A Cas

2.8.2 Kelebihan Modul 1 *Slot Multi Charger*

- *Wiring* yg mudah: cukup 2 kabel ke *battery* (-) dan (+)
- Bentuk kecil hanya 18mm x 20mm dan tebal 12mm
- Dilengkapi berlapis sistem proteksi:



- Proteksi panas berlebihan
- Proteksi *over voltage/over charge* $> 4.2\text{v}$
- Proteksi baterai kosong/*over discharge* $< 2.9\text{v}$
- Proteksi *short circuit* (korslet)
- Support teknologi *trickle mode* dan *zero-voltage charge*

2.8.3 Pemasangan Modul 1 *Slot Multi Charger*

- B+ ke baterai positif (+)
- B- ke baterai negatif (-)
- Tidak boleh terbalik karena akan merusak rangkaian
- Terdapat 2 lampu indikator:
 - Merah mengindikasikan sedang *charging*
 - Biru mengindikasikan sedang pembuangan (*discharger*)

2.8.4 Spesifikasi Modul 1 *Slot Multi Charger*

- *Port Input*: *micro* USB (bisa langsung *disconnect* ke *charger handphone* umum)
- *Port Output*: USB (bisa langsung *disconnect* ke *handphone* umum)
- Tegangan *input*: 3.7~5.5v
- Tegangan *output*: 5v
- Arus *input*: 1A
- Arus *output*: max 1A
- Arus *standby*: 8uA
- Tegangan stop cas penuh: 4.2V +/- 1%
- Tegangan stop *discharge*: 2.9v
- Efisiensi: 85% (*input* 3.7V, *output* 5V 1A)
- Suhu kerja: -10 s/d 85 C

2.9 Buzzer 5V

Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja *buzzer* hampir sama dengan *loud speaker*, jadi *buzzer* juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. *Buzzer* biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (alarm).

Bergetarnya udara yang kemudian menghasilkan bunyi ini bisa disebut sebagai akibat proses resonansi antara getaran yang dihasilkan oleh gerakan kumparan yang menggetarkan diafragma dalam *buzzer* terhadap udara. Resonansi ini akan terpengaruhi seiring perubahan getaran yang terjadi karena perubahan gerakan kumparan yang menggerakkan diafragma akibat perbedaan arus yang diberikan. Dan terpengaruhnya resonansi tersebut, maka frekuensi suara yang dihasilkan oleh *buzzer* pun akan berubah. Hal inilah yang menunjukkan hubungan antara resonansi terhadap frekuensi suara yang dihasilkan. Semakin tinggi resonansi yang terjadi, maka semakin tinggi pula frekuensi suara yang dihasilkan.



Gambar 2.25 Buzzer 5V

(Sumber : abusemark.com/store/images/buzzer.jpg)

2.10 Saklar *Rocker (Power)* Dan *Switch Push Button*

Saklar *rocker* (saklar *power*) adalah perangkat / saklar sederhana yang berfungsi untuk menghubungkan / memutuskan aliran arus listrik dengan sistem kerja dengan cara yang sama seperti saklar *toggle*, yaitu terdapat sebuah tuas yang dapat ditekan ke atas dan ke bawah. Tuas saklar-saklar ini biasanya disebut dengan tuas *rocker*, bentuknya seperti tombol listrik. Umumnya terdapat dua buah angka pada tuas *rocker*, angka 1 dan 0. Apabila tuas ditekan pada angka 1, mengindikasikan bahwa saklar dalam posisi *On*, dan bila tuas ditekan pada angka 0, mengindikasikan saklar dalam posisi *Off*.

Contoh saklar *rocker* seperti yang terlihat dibawah ini:

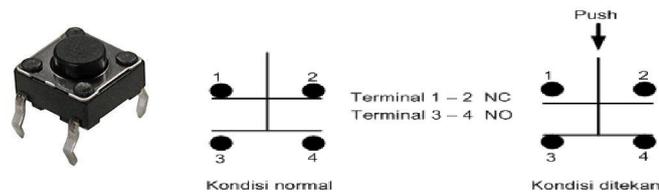


Gambar 2.26 Saklar *Rocker (Power)*

(Sumber : klinikrobot.com/)

Push button switch (saklar tombol tekan) adalah perangkat / saklar sederhana yang berfungsi untuk menghubungkan atau memutuskan aliran arus listrik dengan sistem kerja tekan *unlock* (tidak mengunci). Sistem kerja *unlock* disini berarti saklar akan bekerja sebagai *device* penghubung atau pemutus aliran arus listrik saat tombol ditekan, dan saat tombol tidak ditekan (dilepas), maka saklar akan kembali pada kondisi normal. Sebagai *device* penghubung atau pemutus, *push button switch* hanya memiliki 2 kondisi, yaitu *On* dan *Off* (1 dan 0). Istilah *On* dan *Off* ini menjadi sangat penting karena semua perangkat listrik yang memerlukan sumber energi listrik pasti membutuhkan kondisi *On* dan *Off*.

Prinsip kerja *Push Button* adalah apabila dalam keadaan normal tidak ditekan maka kontak tidak berubah, apabila ditekan maka kontak NC akan berfungsi sebagai stop dan kontak NO akan berfungsi sebagai *start*.

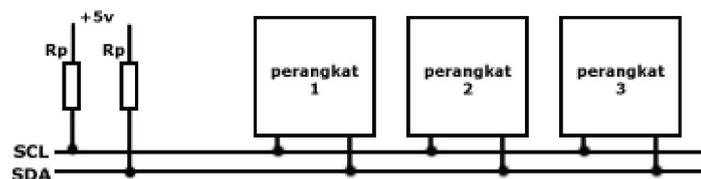
Gambar 2.27 *Switch Push Button* & Prinsip Kerjanya

(Sumber : rizalirfani.blog.st3telkom.ac.id/)

2.11 Komunikasi I2C

Sebenarnya jalur I2C banyak digunakan di PC, telpon selular, radio mobil, TV, sistem terintegrasi di rumah dan perkantoran dan bahkan kartu berganda di *blade server*. Apa sebenarnya yang membuat I2C begitu menarik dan banyak digunakan di perangkat digital? Yaitu, dengan I2C hanya membutuhkan dua jalur untuk berkomunikasi antar perangkat. tidak memerlukan *address decoder* untuk mengimplementasi jalur I2C. Dua jalur tersebut adalah **SDA (Serial Data)** dan **SCL (Serial Clock)**.

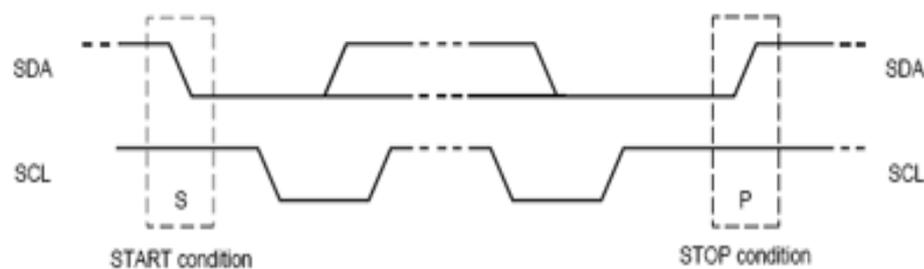
SCL merupakan jalur yang digunakan untuk mensinkronisasi transfer data pada jalur I2C, sedangkan SDA merupakan jalur untuk data. Beberapa perangkat dapat terhubung ke dalam jalur I2C yang sama dimana SCL dan SDA terhubung ke semua perangkat tersebut, hanya ada satu perangkat yang mengontrol SCL yaitu perangkat master. Jalur dari SCL dan SDA ini terhubung dengan *pull-up* resistor yang besar resistansinya tidak menjadi masalah (bisa 1K, 1.8K, 4.7K, 10K, 47K atau nilai diantara *range* tersebut). Kurang lebih gambarnya seperti ini:



Gambar 2.28 Contoh Implementasi Jalur I2C

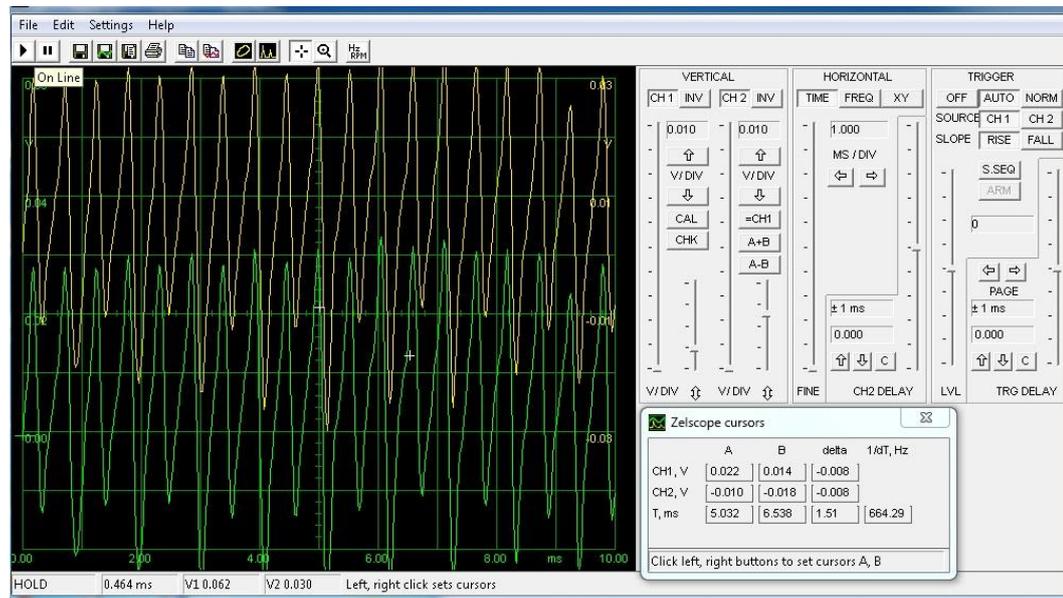
Dengan adanya *pull-up* disini, jalur SCL dan SDA menjadi *open drain*, yang maksudnya adalah perangkat hanya perlu memberikan *output 0 (LOW)* untuk membuat jalur menjadi *LOW*, dan dengan membiarkannya *pull-up* resistor sudah membuatnya *HIGH*. Kembali ke I2C, umumnya dalam I2C ada satu

perangkat yang berperan menjadi *master* (meskipun dimungkinkan beberapa perangkat, dalam jalur I2C yang sama, menjadi *master*) dan satu atau beberapa perangkat *slave*. Dalam jalur I2C, hanya perangkat *master* yang dapat mengontrol jalur SCL yang berarti transfer data harus diinisialisasi terlebih dahulu oleh perangkat *master* melalui serangkaian pulsa *clock* (*slave* tidak bisa, tapi ada satu kasus yang disebut *clock stretching*). Tugas perangkat *slave* hanya merespon apa yang diminta *master*. *Slave* dapat memberi data ke *master* dan menerima data dari *master* setelah *server* melakukan inisialisasi. Misalkan *microcontroller* (uC) adalah perangkat *master* yang terhubung dalam satu I2C dengan perangkat-perangkat *slave* seperti modul pengendali motor servo, modul kompas dan sensor lainnya. Lalu uC dapat mengontrol pergerakan servo dengan memberikan data ke modul servo, mendapatkan data kompas dengan memerintahkan modul kompas agar mengirimkan data. Semua itu hanya dibutuhkan 2 jalur dengan tambahan 2 resistor sebagai *pull-up*. Lalu bagaimana membedakan *slave* tersebut adalah modul servo atau modul kompas? Menggunakan pengalamatan tentunya, dimana setiap perangkat *slave* itu mempunyai alamat yang unik. mengenai sinyal-sinyal yang digunakan dalam protokol I2C. bahwa *master* terlebih dahulu menginisialisasi sebelum memulai transfer data antara *slave*-nya. Inisialisasi diawali dengan sinyal *START* (transisi *high* ke *low* pada jalur SDA dan kondisi *high* pada jalur SCL, lambang S pada gambar 2), lalu transfer data dan sinyal *STOP* (transisi *low* ke *high* pada jalur SDA dan kondisi *high* pada jalur SCL, lambang P pada gambar 2.29) untuk menandakan akhir transfer data.



Gambar 2.29 Sinyal untuk *START* dan *STOP*

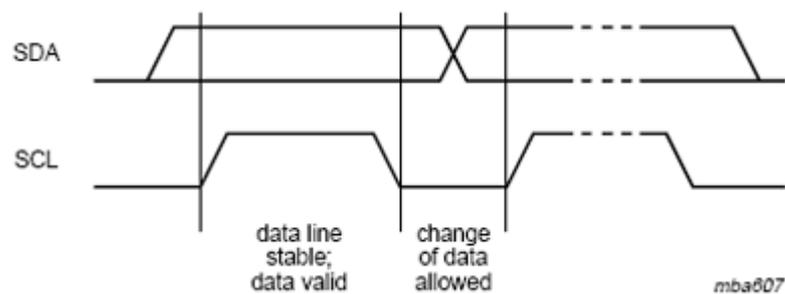
(Sumber: UM10204 I2C-bus specification and user manual)



Gambar 2.30 Grafik Sinyal *START* dan *STOP* (kuning sda, hijau scl)

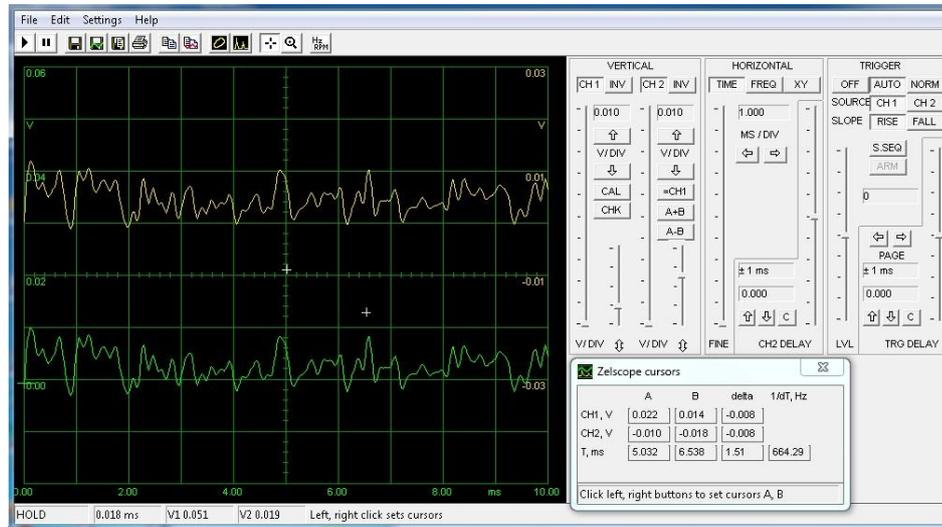
(Sumber : Dikelola Sendiri)

Banyaknya byte yang dapat dikirimkan dalam satu transfer data itu tidak ada aturannya. Jika transfer data yang ingin dilakukan sebesar 2 byte, maka pengiriman pertama adalah 1 byte dan setelah itu 1 byte. Setiap byte yang di transfer harus diikuti dengan bit *Acknowledge* (ACK) dari penerima, menandakan data berhasil diterima. Byte yang dikirim dari pengirim diawali dari bit MSB. Saat bit dikirim, pulsa *clock* (SCL) di set ke *HIGH* lalu ke *LOW*. Bit yang dikirim pada jalur SDA tersebut harus stabil saat periode *clock* (SCL) *HIGH*. Kondisi *HIGH* atau *LOW* dari jalur data (SDA) hanya dapat berubah saat kondisi sinyal SCL itu *LOW*.



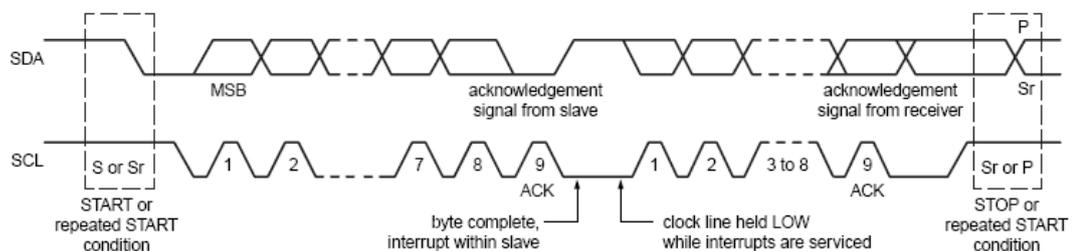
Gambar 2.31 Transfer Bit pada Jalur I2C

(Sumber: UM10204 I2C-bus specification and user manual)

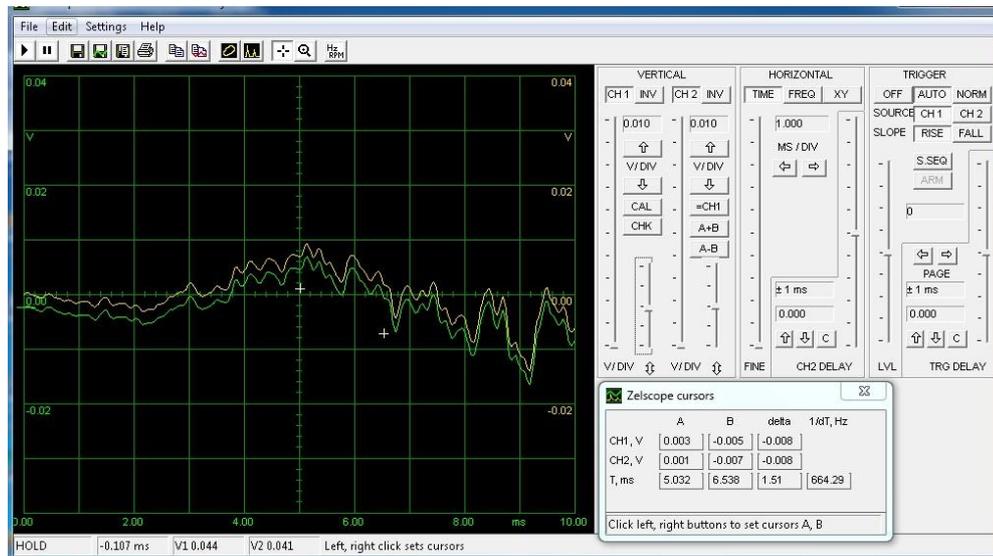


Gambar 2.32 Grafik Transfer Bit pada Jalur I2C (kuning sda, hijau scl)
(Sumber : Dikelola Sendiri)

Setiap pulsa *clock* itu dihasilkan (di jalur SCL) untuk setiap bit (di jalur SDA) yang ditransfer. Jadi untuk pengiriman 8 bit akan ada 9 pulsa *clock* yang harus dihasilkan (1 lagi untuk bit ACK). Kronologi sebelum perangkat penerima memberikan sinyal ACK adalah sebagai berikut: saat pengirim selesai mengirimkan bit ke-8, pengirim melepaskan jalur SDA ke *pull-up* (ingat penjelasan *open drain* di atas) sehingga menjadi *HIGH*. Nah saat kondisi tersebut terjadi, penerima harus memberikan kondisi *LOW* ke SDA saat pulsa *clock* ke-9 berada dalam kondisi *HIGH*.



Gambar 2.33 Transfer Data (Byte) pada Jalur I2C
(sumber: UM10204 I2C-bus specification and user manual)



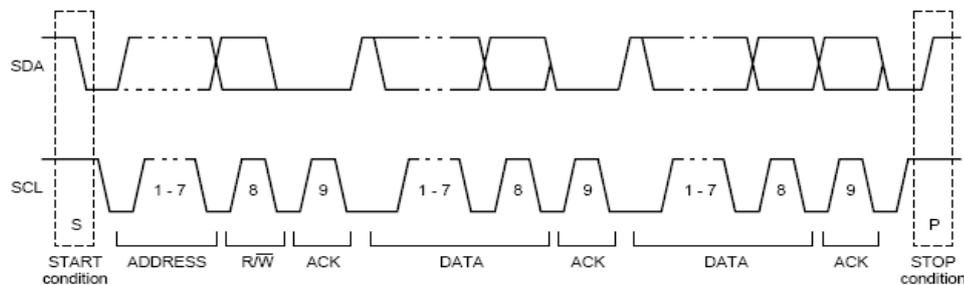
Gambar 2.34 Grafik Transfer Data (Byte) pada Jalur I2C (kuning sda, hijau scl)
(Sumber : Dikelola Sendiri)

Jika SDA tetap dalam kondisi *HIGH* saat pulsa *clock* ke-9, maka ini didefinisikan sebagai sinyal *Not Acknowledge* (NACK). *Master* dapat menghasilkan sinyal *STOP* untuk mensudahi transfer, atau mengulang sinyal *START* untuk memulai transfer data yang baru. Ada 5 kondisi yang menyebabkan NACK:

1. Tidak adanya penerima dengan alamat yang diminta pada jalur, sehingga tidak ada perangkat yang merespon ACK.
2. Penerima tidak dapat menerima atau mengirim karena sedang mengeksekusi fungsi lain dan tidak siap untuk memulai komunikasi dengan *master*.
3. Pada saat transfer data, penerima mendapatkan data atau perintah yang tidak dimengerti oleh penerima.
4. Pada saat transfer data, penerima tidak dapat menerima lagi byte data yang dikirimkan.
5. Penerima-*master* perlu memberi sinyal pengakhiran transfer data ke penerima-*slave*.

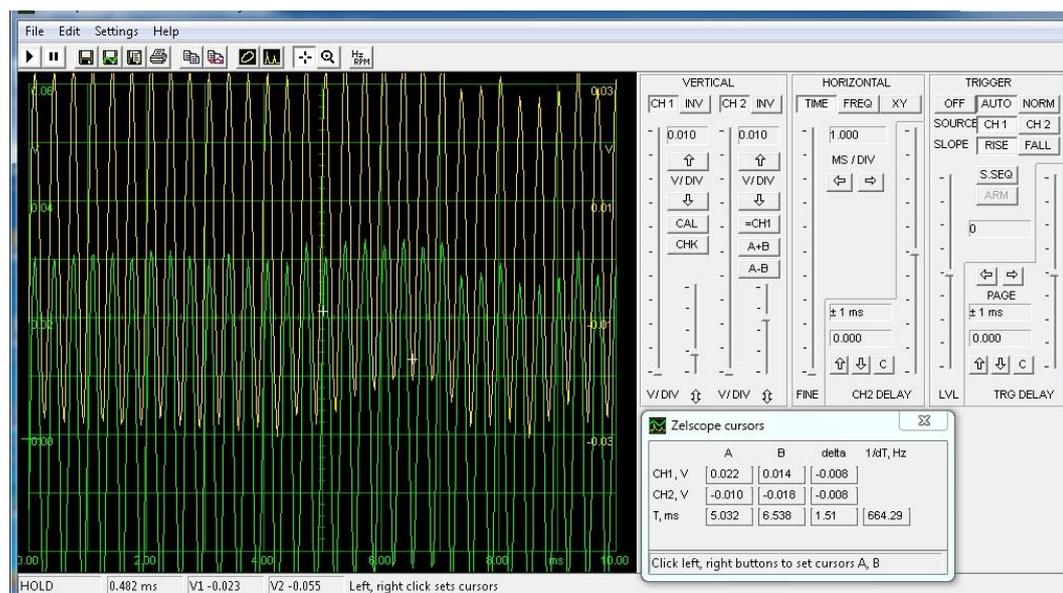
Pengalamatan dalam I2C bisa 7 bit atau 10 bit. Pengalamatan 10 bit jarang digunakan dan juga tidak dibahas di sini. Semua perangkat (uC dan modul-modul)

yang terhubung ke dalam jalur I2C yang sama dapat dialamati sebanyak 7 bit. Ini berarti sebuah jalur I2C dengan pengalamatan 7 bit dapat menampung 128 (2^7) perangkat. Saat mengirimkan data alamat (yang 7 bit itu), kita tetap mengirim data 1 byte (8 bit). 1 bit lagi digunakan untuk menginformasikan perangkat *slave* apakah *master* menulis (*write*) data ke *slave* atau membaca (*read*) data dari *slave*. Jika bit tersebut 0, maka *master* menulis data ke *slave*. Jika bit tersebut 1, maka *master* membaca data dari *slave*. Bit ini (untuk informasi tulis/baca) merupakan LSB, sedangkan sisanya adalah data alamat 7 bit. Berikut adalah contoh sinyal yang dimulai dengan data alamat lalu data yang ingin ditransfer ke alamat tersebut:



Gambar 2.35 Sinyal Alamat dan Data

(Sumber: UM10204 I2C-bus specification and user manual)



Gambar 2.36 Grafik Sinyal Alamat dan Data

(Sumber : Dikelola Sendiri)



Sebaiknya saat mengalami perangkat-perangkat dalam I2C anggap saja menggunakan 8-bit. Jika menggunakan 7 bit justru akan membingungkan. Misal diberikan alamat 0×14 (dalam penghitungan 7 bit), maka untuk menulis ke alamat 0×14 kita harus memberikan byte 0×28 dengan menggesernya 1 bit (bit 0 pada LSB berarti menulis). Contoh yang memudahkan adalah HMC5883L yang memiliki alamat $0 \times C0$ (perhitungan 8 bit). Untuk menulis ke HMC5883L kita menggunakan $0 \times C0$ dan untuk membaca dari HMC5883L kita menggunakan $0 \times C1$.

Dalam konsep bus I2C ada beberapa istilah yang didefinisikan sebagai berikut :

- Komponen yang mengirim sinyal ke bus I2C disebut sebagai *transmitter*
- Komponen yang menerima sinyal disebut sebagai *receiver*
- Komponen yang mengontrol transfer sinyal pada bus I2C dengan frekuensi detak tertentu disebut sebagai *master*
- Komponen yang dikontrol oleh *master* disebut sebagai *slave*

Dalam konsep komunikasi bus I2C hanya dibutuhkan enam operasi dasar sederhana untuk mentransmisikan dan menerima informasi. Keenam kode operasi itu adalah sebagai berikut :

1. Sebuah bit *start*
2. Suatu alamat *slave* 7-bit
3. Suatu bit *read/write* yang akan menentukan apakah *slave* akan berfungsi sebagai *transmitter* atau *receiver*.
4. Suatu bit pemberitahuan (*acknowledge*)
5. Bit pesan yang dibagi ke dalam segmen 8-bit
6. Suatu bit *stop*

Fitur utama I2C bus adalah sebagai berikut :

- Hanya melibatkan dua kabel yaitu *serial data line* (selanjutnya disebut SDA) dan *serial clock line* (selanjutnya disebut SCL).



-
- Setiap IC yang terhubung dalam I2C memiliki alamat yang unik yang dapat diakses secara *software* dengan *master/slave protocol* yang sederhana, dan mampu mengakomodasikan multi *master*.
 - I2C merupakan serial bus dengan orientasi data 8 bit (byte), komunikasi 2 arah, dengan kecepatan transfer data sampai 100Kbit/s pada *mode* standar dan 3,4 Mbit/s pada mode kecepatan tinggi.
 - Jumlah IC yang dapat dihubungkan pada I2C bus hanya dibatasi oleh beban kapasitansi pada bus yaitu maksimum 400pF.

Keuntungan yang didapat dengan menggunakan I2C antara lain :

- Meminimalkan jalur hubungan antar IC (bandingkan dengan *parallel bus*).
- Menghemat luasan PCB yang dibutuhkan.
- Membuat sistem yang didesain berorientasi *software* (mudah diekspan dan *diupgrade*).
- Membuat sistem yang didesain menjadi standar, sehingga dapat dihubungkan dengan sistem lain yang juga menggunakan I2C bus.