

BAB II

TINJAUAN PUSTAKA

2.1 Mesin CNC

Awal lahirnya mesin CNC (Computer Numerically Controlled) bermula dari 1952 yang dikembangkan oleh John Pearson dari Institut Teknologi Massachusetts, atas nama Angkatan Udara Amerika Serikat. Semula proyek tersebut diperuntukkan untuk membuat benda kerja khusus yang rumit. Semula perangkat mesin CNC memerlukan biaya yang tinggi dan volume unit pengendali yang besar.

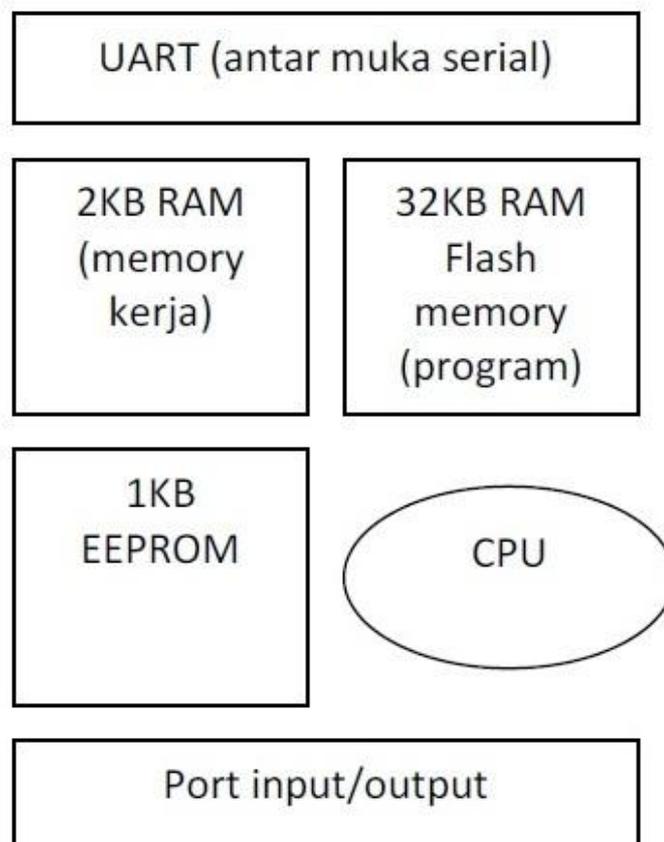
Pada tahun 1973, mesin CNC masih sangat mahal sehingga masih sedikit perusahaan yang mempunyai keberanian dalam memelopori investasi dalam teknologi ini. Dari tahun 1975, produksi mesin CNC mulai berkembang pesat. Perkembangan ini dipacu oleh perkembangan mikroprosesor, sehingga volume unit pengendali dapat lebih ringkas (Labinovasi, 2013).

Mesin CNC merupakan mesin yang dikontrol oleh komputer dengan menggunakan bahasa numerik (data perintah dengan kode angka, huruf dan simbol) sesuai standar *ISO*. Mesin CNC ini menggunakan *software-software* antara lain Arduino, Processing, Camotic, dan Inkscape. Adapun beberapa *hardware*-nya yaitu Arduino Uno, Adafruit Motor Shield, Nema 16 Stepping Motor, dan mekaniknya. Proses perancangan harus dikerjakan dengan tepat, mulai dari pemilihan bahan untuk mekanik, perancangan mesin, peng-installan *software*, penyingkronan hingga pengujian, agar mesin dapat bekerja sesuai dengan program yang telah dibuat dan menghasilkan produk yang diinginkan.

2.2 Mikrokontroler Arduino Uno

Mikrokontroler adalah sebuah chip yang berfungsi sebagai pengontrol rangkaian elektronik dan umumnya dapat menyimpan program didalamnya. Mikrokontroler umumnya terdiri dari CPU (Central Processing Unit), memori, I/O tertentu dan unit pendukung seperti Analog-to-Digital Converter (ADC) yang sudah terintegrasi di dalamnya (Elektronika, 2012).

Arduino adalah kit elektronik atau papan rangkaian elektronik open source yang di dalamnya terdapat komponen utama yaitu sebuah chip mikrokontroler dengan jenis AVR dari perusahaan Atmel (Santosa, 2012). Chip atau IC (integrated circuit) dapat diprogram menggunakan komputer. Tujuan menanamkan program pada mikrokontroler adalah agar rangkaian elektronik dapat membaca input, memproses input tersebut dan kemudian menghasilkan output sesuai yang diinginkan. Jadi mikrokontroler bertugas sebagai ‘otak’ yang mengendalikan input, proses dan output sebuah rangkaian elektronik (Adi Putra, 2014). Diagram Blok Arduino dapat dilihat pada gambar 2.1 .



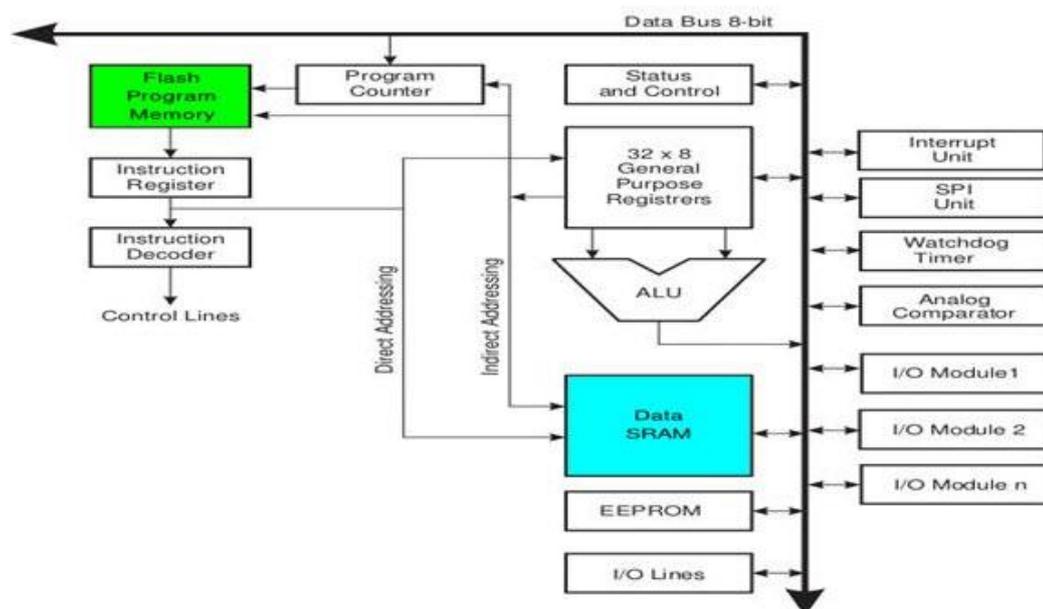
Gambar 2.1 Diagram Blok Arduino

Dalam Diagram Blok Arduino terdiri dari beberapa bagian, yaitu :

- a. *Universal Asynchronous Receiver/Transmitter* (UART) adalah antar muka yang digunakan untuk komunikasi serial seperti pada RS-232, RS-422 dan RS-485.

- b. 2KB RAM pada memory kerja bersifat *volatile* (hilang saat daya dimatikan), digunakan oleh variable-variabel di dalam program.
- c. 32KB RAM flash memory bersifat *non-volatile*, digunakan untuk menyimpan program yang dimuat dari komputer. Selain program, flash memory juga menyimpan *bootloader*.
- d. *Bootloader* adalah program inisiasi yang ukurannya kecil, dijalankan oleh CPU saat daya dihidupkan. Setelah *bootloader* selesai dijalankan, berikutnya program di dalam RAM akan dieksekusi.
- e. 1KB EEPROM bersifat *non-volatile*, digunakan untuk menyimpan data yang tidak boleh hilang saat daya dimatikan. Tidak digunakan pada papan Arduino (*red*: namun bisa diakses/diprogram oleh pemakai dan digunakan sesuai kebutuhan).
- f. Central Processing Unit (CPU), bagian dari microcontroller untuk menjalankan setiap instruksi dari program.
- g. Port input/output, pin-pin untuk menerima data (input) digital atau analog, dan mengeluarkan data (output) digital atau analog.

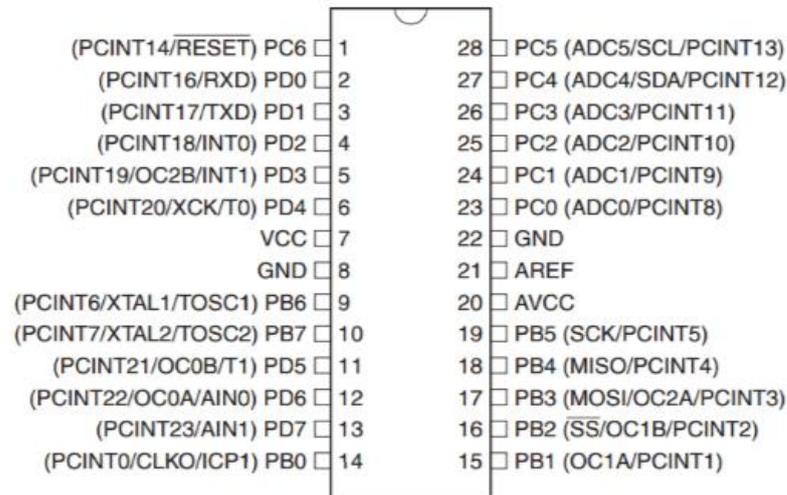
Untuk lebih jelasnya tentang Architecture arduino uno bisa di lihat pada Gambar 2.2 .



Gambar 2.2 Architecture Atmega 328

2.2.1 Konfigurasi Pin Atmega328

Berikut adalah Konfigurasi ATmega328 dapat dilihat pada gambar 2.3 .



Gambar 2.3 Konfigurasi Pin Atmega 328

ATMega328 memiliki 28 Pin, yang masing-masing pinnya memiliki fungsi yang berbeda-beda baik sebagai port maupun fungsi yang lainnya. Berikut akan dijelaskan fungsi dari masing-masing kaki ATmega8 yaitu sebagai berikut :

- VCC Merupakan supply tegangan digital.
- GND Merupakan ground untuk semua komponen yang membutuhkan grounding.
- Port B (PB7-PB0) Didalam Port B terdapat XTAL1, XTAL2, TOSC1, TOSC2. Jumlah Port B adalah 8 buah pin, mulai dari pin B.0 sampai dengan B.7. Tiap pin dapat digunakan sebagai input maupun output. Port B merupakan sebuah 8-bit bidirectional I/O dengan internal pull-up resistor. Sebagai input, pin-pinyang terdapat pada port B yang secara eksternal diturunkan, maka akan mengeluarkan arus jika pull-up resistor diaktifkan. Khusus PB6 dapat digunakan sebagai input Kristal (inverting oscillator amplifier) dan input ke rangkaian clock internal, bergantung pada pengaturan Fuse bit yang digunakan untuk memilih sumber clock. Sedangkan untuk PB7 dapat digunakan sebagai output Kristal (output

oscillator amplifier) bergantung pada pengaturan Fuse bit yang digunakan untuk memilih sumber clock. Jika sumber clock yang dipilih dari oscillator internal, PB7 dan PB6 dapat digunakan sebagai I/O atau jika menggunakan Asynchronous Timer/Counter2 maka PB6 dan PB7 (TOSC2 dan TOSC1) digunakan untuk saluran input timer.

- Port C (PC5-PC0) Port C merupakan sebuah 7-bit bi-directional I/O port yang di dalam masing-masing pin terdapat pull-up resistor. Jumlah pin nya hanya 7 buah mulai dari pin C.0 sampai dengan pin C.6. Sebagai keluaran/output port C memiliki karakteristik yang sama dalam hal menyerap arus (sink) ataupun mengeluarkan arus (source).
- RESET/PC6 Jika RSTDISBL Fuse diprogram, maka PC6 akan berfungsi sebagai pin I/O. Pin ini memiliki karakteristik yang berbeda dengan pin-pin yang terdapat pada port C lainnya. Namun jika RSTDISBL Fuse tidak diprogram, maka pin ini akan berfungsi sebagai input reset. Dan jika level tegangan yang masuk ke pin ini rendah dan pulsa yang ada lebih pendek dari pulsaminimum, maka akan menghasilkan suatu kondisi reset meskipun clock-nya tidak bekerja.
- Port D (PD7-PD0) Port D merupakan 8-bit bi-directional I/O dengan internal pull-up resistor. Fungsi dari port ini sama dengan port-port yang lain. Hanya saja pada port ini tidak terdapat kegunaan-kegunaan yang lain. Pada port ini hanya berfungsi sebagai masukan dan keluaran saja atau biasa disebut dengan I/O.
- Avcc Pin ini berfungsi sebagai supply tegangan untuk ADC. Untuk pin ini harus dihubungkan secara terpisah dengan VCC karena pin ini digunakan untuk analog saja. Bahkan jika ADC pada AVR tidak digunakan tetap saja disarankan untuk menghubungkannya secara terpisah dengan VCC. Jika ADC digunakan, maka AVcc harus dihubungkan ke VCC melalui low pass filter.
- AREF Merupakan pin referensi jika menggunakan ADC

Arduino Uno dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber listrik dipilih secara otomatis. Eksternal (non-USB) daya dapat datang baik dari AC-DC adaptor atau baterai. Adaptor ini dapat dihubungkan dengan cara menghubungkannya *plug* pusat-positif 2.1 mm ke dalam board colokan listrik. Lead dari baterai dapat dimasukkan ke dalam *header* pin Gnd dan Vin dari konektor *Power. Board* dapat beroperasi pada pasokan daya dari 6 - 20 volt. Jika diberikan dengan kurang dari 7V, bagaimanapun, pin 5V dapat menyuplai kurang dari 5 volt dan *board* mungkin tidak stabil. Jika menggunakan lebih dari 12V, regulator tegangan bisa panas dan merusak *board*. Rentang yang dianjurkan adalah 7 - 12 volt.

Pin catu daya adalah sebagai berikut:

- VIN. Tegangan input ke *board* Arduino ketika menggunakan sumber daya eksternal (sebagai lawan dari 5 volt dari koneksi USB atau sumber daya lainnya diatur). Anda dapat menyediakan tegangan melalui pin ini, atau, jika memasok tegangan melalui colokan listrik, mengaksesnya melalui pin ini.
- 5V. Catu daya diatur digunakan untuk daya mikrokontroler dan komponen lainnya di *board*. Hal ini dapat terjadi baik dari VIN melalui regulator onboard, atau diberikan oleh USB .
- 3,3 volt pasokan yang dihasilkan oleh regulator on-board. Menarik arus maksimum adalah 50 mA.
- GND

2.2.2 Memory

ATmega328 ini memiliki 32 KB dengan 0,5 KB digunakan untuk *loading file*. Ia juga memiliki 2 KB dari SRAM dan 1 KB dari EEPROM

2.2.3 Input & Output

Masing-masing dari 14 pin digital pada Uno dapat digunakan sebagai input atau output, menggunakan fungsi `pinMode()`, `digitalWrite()`, dan

`digitalRead()`. Mereka beroperasi di 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor pull-up internal dari 20-50 K Ω .

Selain itu, beberapa pin memiliki fungsi khusus:

- Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data TTL serial. Pin ini terhubung ke pin yang sesuai dari chip ATmega8U2 USB-to-Serial TTL.
- Eksternal Interupsi: 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu interupsi pada nilai yang rendah, tepi naik atau jatuh, atau perubahan nilai. Lihat `attachInterrupt ()` fungsi untuk rincian.
- PWM: 3, 5, 6, 9, 10, dan 11. Menyediakan 8-bit output PWM dengan `analogWrite ()` fungsi.
- SPI: 10 (SS), 11 (mosi), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI.
- LED: 13. Ada built-in LED terhubung ke pin digital 13. Ketika pin adalah nilai TINGGI, LED menyala, ketika pin adalah RENDAH, itu off.

Uno memiliki 6 input analog, diberi label A0 melalui A5, masing-masing menyediakan 10 bit resolusi yaitu 1024 nilai yang berbeda. Secara default sistem mengukur dari tanah sampai 5 volt.

- TWI: A4 atau SDA pin dan A5 atau SCL pin. Mendukung komunikasi TWI
- Aref. Referensi tegangan untuk input analog. Digunakan dengan `analogReference ()`.
- Reset.

Lihat juga pemetaan antara pin Arduino dan ATmega328 port. Pemetaan untuk ATmega8, 168 dan 328 adalah identik.

2.2.4 Komunikasi

Uno Arduino memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lain. ATmega328 ini menyediakan UART TTL (5V) komunikasi serial, yang tersedia pada pin digital 0 (RX) dan 1 (TX). Sebuah ATmega16U2 pada saluran *board* ini komunikasi serial melalui

USB dan muncul sebagai com port virtual untuk perangkat lunak pada komputer. *Firmware* Arduino menggunakan USB *driver* standar COM, dan tidak ada *driver* eksternal yang dibutuhkan. Namun, pada Windows, file. Inf diperlukan. Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data sederhana yang akan dikirim ke *board* Arduino. RX dan TX LED di *board* akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dan koneksi USB ke komputer. ATmega328 ini juga mendukung komunikasi I2C (TWI) dan SPI. Fungsi ini digunakan untuk melakukan komunikasi inteface pada sistem.

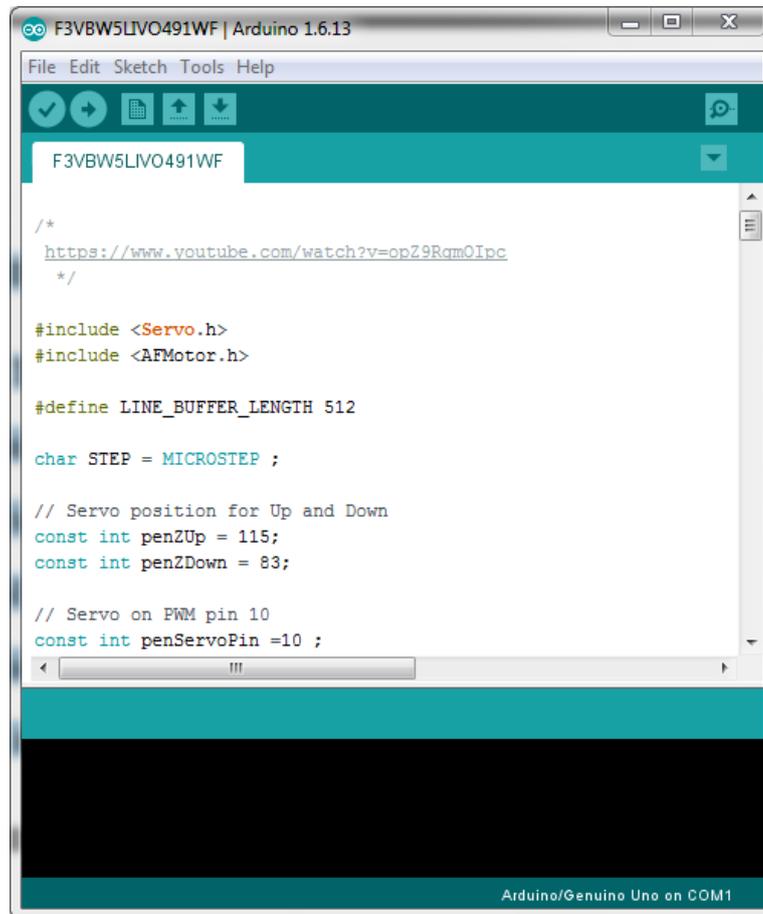
2.2.5 Programming

Uno Arduino dapat diprogram dengan perangkat lunak Arduino. Pilih Arduino Uno dari *Tool* lalu sesuaikan dengan mikrokontroler yang digunakan. Para ATmega328 pada *Uno Arduino* memiliki bootloader yang memungkinkan Anda untuk meng-upload program baru untuk itu tanpa menggunakan programmer hardware eksternal. Ini berkomunikasi menggunakan protokol dari bahas C. Sistem dapat menggunakan perangkat lunak FLIP Atmel (Windows) atau programmer DFU (Mac OS X dan Linux) untuk memuat *firmware* baru. Atau Anda dapat menggunakan header ISP dengan programmer eksternal.

2.2.6 Perangkat Lunak (Arduino IDE)

IDE itu merupakan kependekan dari *Integrated Development Enviroenment*. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Di dalam mikrokontroler Arduino telah di masukkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler.

Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan library C/C++ yang biasa disebut *Wiring* yang membuat operasi *input* dan *output* menjadi lebih mudah. Arduino IDE ini dikembangkan dari software *Processing* yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino.



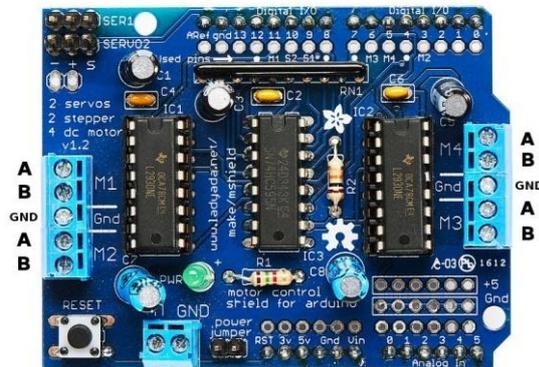
Gambar 2.4 Tampilan Program *Arduino*

2.2.7 Otomatis Software Reset

Tombol reset *Uno Arduino* dirancang untuk menjalankan program yang tersimpan didalam mikrokontroller dari awal. Tombol reset terhubung ke Atmega328 melalui kapasitor 100nf. Setelah tombol reset ditekan cukup lama untuk me-reset chip, *software IDE* *Arduino* dapat juga berfungsi untuk meng-*upload* program dengan hanya menekan tombol *upload* di *software IDE* *Arduino*.

2.3 Adafruit Motor Shield

Adafruit Motor Shield merupakan shield arduino yang mudah penggunaannya untuk pembuatan aplikasi robot. Karena shield ini dapat menjalankan empat buah motor dan dua buah servo sekaligus. Bentuk fisik driver adafruit motor shield dapat dilihat pada gambar 2.5 .

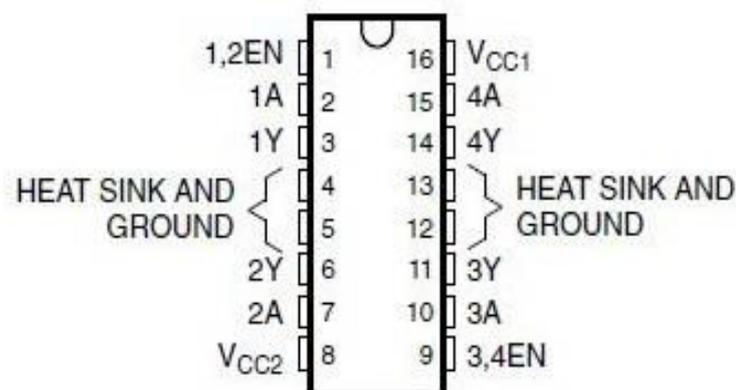


Gambar 2.5 Adafruit Motor Shield

Spesifikasi lengkap shield ini adalah sebagai berikut.

- a. 2 konektor untuk 5V Servo.
- b. Dapat menjalankan 4 motor DC atau 2 stepper motor atau 2 Servo.
- c. Dapat menjalankan 4 motor bi-directional DC dengan kecepatan pemilihan 8-bit.
- d. Menjalankan 2 stepper motor (unipolar atau bipolar) dengan single coil atau double coil.
- e. 4 H-Bridges: per bridge menyediakan 0.6A (1.2A saat puncak) dengan
- f. perlindungan termal, dapat menjalankan motor 4.5V sampai 36V DC.
- g. Tombol reset.
- h. 2 konektor daya eksternal.

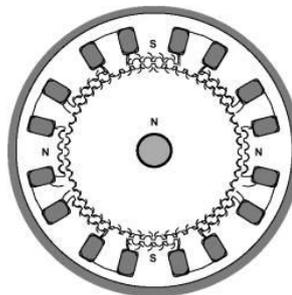
Untuk lebih jelasnya mengenai konfigurasi ic l293d dapat di lihat pada gambar 2.6 .



Gambar 2.6 Ic L293D

2.4 Motor Stepper

Motor stepper adalah motor yang digunakan sebagai penggerak atau pemutar. Prinsip kerja motor stepper mirip dengan motor DC, sama-sama dicatu dengan tegangan DC untuk memperoleh medan magnet. Bila motor DC memiliki magnet tetap pada stator, motor stepper mempunyai magnet tetap pada rotor. Motor stepper dinyatakan dengan spesifikasi : “berapa fasa “, “berapa derajat perstep”, “berapa volt tegangan catu untuk tiap lilitan” dan ”berapa ampere/miliampere arus yang dibutuhkan untuk tiap lilitan”. Motor stepper tidak dapat bergerak sendirinya, tetapi bergerak secara per-step sesuai dengan spesifikasinya, dan bergerak dari satu step ke step berikutnya memerlukan waktu, serta menghasilkan torsi yang besar pada kecepatan rendah. Motor stepper juga memiliki karakteristik yang lain yaitu torsi penahan, yang memungkinkan menahan posisinya. Hal ini sangat berguna untuk aplikasi dimana suatu sistem memerlukan keadaan start dan stop (Trianto, 2005). Bagian dalam Motor stepper dapat dilihat pada gambar 2.7 .



Gambar 2.7 Motor Stepper

Motor stepper tidak merespon sinyal clock dan mempunyai beberapa lilitan dimana lilitan-lilitan tersebut harus dicatu (tegangan) dahulu dengan suatu urutan tertentu agar dapat berotasi. Membalik urutan pemberian tegangan tersebut akan menyebabkan putaran motor stepper yang berbalik arah. Jika sinyal kontrol tidak terkirim sesuai dengan perintah maka motor stepper tidak akan berputar secara tepat, mungkin hanya akan bergetar dan tidak bergerak. Untuk mengontrol motor stepper digunakan suatu rangkaian driver yang menangani kebutuhan arus dan tegangan (Trianto, 2005).

2.4.1 Karakteristik Motor Stepper

Karakteristik dari motor stepper menurut Trianto adalah sebagai berikut:

- a. Tiap motor stepper mempunyai tegangan rata-rata yang tertulis pada tiap unitnya atau tercantum pada datasheet masing-masing motor stepper. Tegangan rata-rata ini harus diperhatikan dengan seksama karena bila melebihi dari tegangan rata-rata ini akan menimbulkan panas yang menyebabkan kinerja putarannya tidak maksimal atau bahkan motor stepper akan rusak dengan sendirinya.
- b. Resistansi per lilitan adalah karakteristik yang lain dari motor stepper. Resistansi ini akan menentukan arus yang mengalir, selain itu juga akan mempengaruhi torsi dan kecepatan maksimum dan motor stepper.
- c. Derajat per step adalah faktor terpenting dalam pemilihan motor stepper sesuai dengan aplikasinya. Tiap-tiap motor stepper mempunyai spesifikasi masing-masing, antara lain: 0.72° per step, 1.8° per step, 3.6° per step, 7.5° per step, 15° per step, dan bahkan ada yang 90° per step. Dalam pengoperasiannya kita dapat menggunakan 2 prinsip yaitu full step atau half step. Dengan full step berarti motor stepper berputar sesuai dengan spesifikasi derajat per stepnya, sedangkan half step berarti motor stepper berputar setengah derajat per step dari spesifikasi motor stepper tersebut.

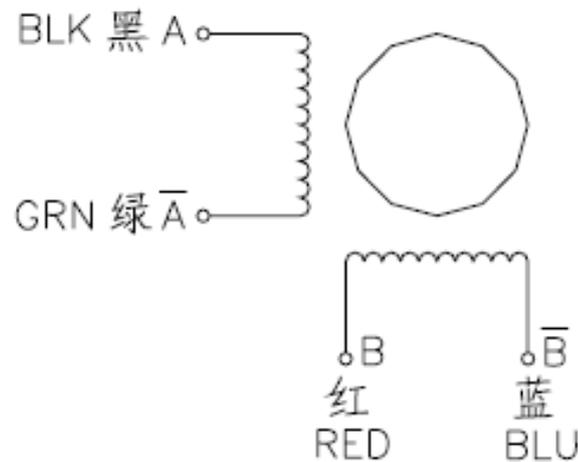
2.4.2 Jenis Motor Stepper

Motor stepper dibedakan menjadi dua kategori besar yaitu: magnet permanen dan reluktansi variabel. Tipe magnet permanen terbagi menjadi dua motor stepper yaitu motor stepper unipolar dan bipolar.

Motor stepper unipolar sangat mudah untuk dikontrol dengan menggunakan rangkaian counter '-n'. Motor stepper unipolar mempunyai karakteristik khusus yaitu berupa lilitan center-tapped dan 1 lilitan sebagai common. Lilitan common akan mencatu tegangan pada center-tapped dan sebagai ground adalah rangkaian drivernya.

Motor stepper unipolar dapat dikenali dengan mengetahui adanya lilitan center-tapped. Jumlah phase dan motor stepper adalah dua kali dan jumlah

koilnya. Umumnya pada motor stepper unipolar terdapat dua buah koil (Trianto, 2005). Diagram motor stepper dapat dilihat pada gambar 2.8 .



Gambar 2.8 Diagram Motor Stepper 4 Wire

2.4.3 Perisip Kerja Motor Stepper

Pada prinsipnya ada dua macam cara kerja motor stepper unipolar, yaitu full-step dan half-step yang di jelaskan pada Tabel 2.1 dan Tabel 2.2

Tabel 2.1 Metode FullStep

| FULLSTEP | | | | | | | | |
|--------------------------------------|-----------------------------|----|----|----|------------------------------|----|----|----|
| Tegangan yang diberikan pada lilitan | | | | | | | | |
| No | Arah putar searah jarum jam | | | | Arah putar melawan jarum jam | | | |
| | L3 | L2 | L1 | L0 | L3 | L2 | L1 | L0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 4 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

Pada full step, suatu titik pada sebuah kutub magnet di rotor akan kembali mendapat tarikan medan magnet stator pada lilitan yang sama setelah step ke 4., dan berikutnya dapat diberikan lagi mulai dari step 1. Setiap step, rotor bergerak searah atau berlawanan dengan jarum jam sebesar spesifikasi derajat per step dan

motor stepper. Setiap step hanya menarik sebuah kutub saja. Tegangan ‘1’ adalah menunjukkan logika dalam level Transistor Transistor Logic (TTL). Besar tegangan sesungguhnya diatur dengan spesifikasi motor stepper yang dipakai, misalnya dengan menggunakan buffer.

Tabel 2.2 Metode HalfStep

| HALFSTEP | | | | | | | | |
|--------------------------------------|-----------------------------|----|----|----|------------------------------|----|----|----|
| Tegangan yang diberikan pada lilitan | | | | | | | | |
| No | Arah putar searah jarum jam | | | | Arah putar melawan jarum jam | | | |
| | L3 | L2 | L1 | L0 | L3 | L2 | L1 | L0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 2 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 3 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 4 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 5 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 6 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 7 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 8 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

Untuk half step, setiap kutub magnet pada rotor akan kembali mendapatkan tarikan dan medan magnet lilitan yang sama setelah step ke 8. berikutnya kembali mulai step 1. Setiap step posisi rotor berubah sebesar setengah derajat dan spesifikasi derajat per step motor stepper.

Berbeda dengan motor stepper unipolar, motor stepper bipolar sangat sulit dalam pengontrolannya. Motor stepper jenis ini memerlukan rangkaian driver yang kompleks. Keuntungan motor stepper bipolar adalah ukurannya yang besar dan dapat menghasilkan torsi yang besar daripada motor stepper unipolar. Motor stepper bipolar di desain dengan koil yang terpisah yang akan di catu dan dua arah (polaritas harus dibalik selama pencatuan). Motor stepper bipolar menggunakan logika yang sama seperti motor stepper unipolar yaitu hanya ‘0’ dan ‘1’ untuk merespon koilnya.

2.5 Motor Servo

Motor servo adalah sebuah perangkat atau aktuator putar (motor) yang dirancang dengan sistem kontrol umpan balik loop tertutup (servo), sehingga dapat di set-up atau di atur untuk menentukan dan memastikan posisi sudut dari poros output motor.

Motor servo adalah jenis motor yang memiliki tiga kabel. Masing-masing digunakan sebagai catu daya, ground, dan kontrol.

Kabel kontrol digunakan untuk menentukan motor untuk memutar rotor ke arah posisi tertentu. Biasanya, rotor hanya berputar hingga 200° . Namun, ada pula yang mampu berputar sebesar 360° .

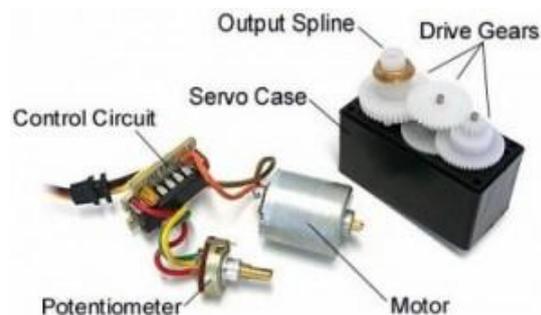
Motor servo biasa digunakan untuk menggerakkan lengan robot atau memutar pada alat ukur yang bersifat analog, atau sebagai aksesoris tambahan proyek Arduino.

Biasanya, motor servo ini dilengkapi dengan tiga kabel berwarna merah, oranye, dan coklat.

- a. Kabel merah dihubungkan ke catu daya 5 volt.
- b. Kabel oranye dihubungkan ke pin digital yang mendukung PWM.
- c. Kabel coklat dihubungkan ke ground.

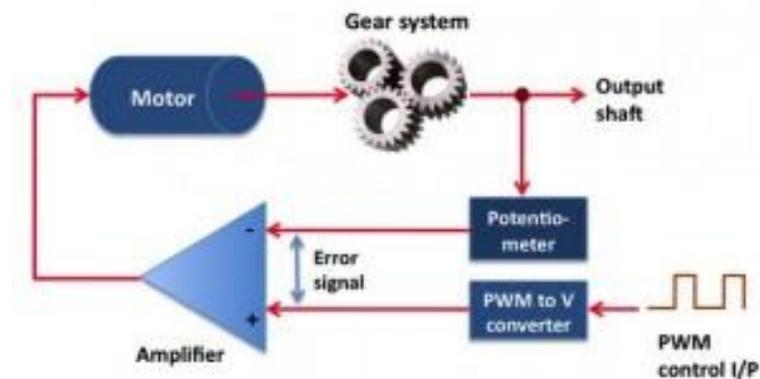
2.5.1 Perisip Kerja Motor Servo

Motor servo dikendalikan dengan memberikan sinyal modulasi lebar pulsa (Pulse Wide Modulation / PWM) melalui kabel kontrol. Lebar pulsa sinyal kontrol yang diberikan akan menentukan posisi sudut putaran dari poros motor servo. Komponen motor servo dapat dilihat pada gambar 2.9 .



Gambar 2.9 Komponen Motor Servo

Sebagai contoh, lebar pulsa dengan waktu 1,5 ms (mili detik) akan memutar poros motor servo ke posisi sudut 90° . Bila pulsa lebih pendek dari 1,5 ms maka akan berputar ke arah posisi 0° atau ke kiri (berlawanan dengan arah jarum jam), sedangkan bila pulsa yang diberikan lebih lama dari 1,5 ms maka poros motor servo akan berputar ke arah posisi 180° atau ke kanan (searah jarum jam), diagram blok motor servo bisa di lihat pada gambar 2.10.



Gambar 2.10 Diagram Blok Motor Servo

2.6 Komunikasi Serial

Komunikasi serial adalah komunikasi yang pengiriman datanya per-bit secara berurutan dan bergantian. Komunikasi ini mempunyai suatu kelebihan yaitu hanya membutuhkan satu jalur dan kabel yang sedikit dibandingkan dengan komunikasi paralel. Pada prinsipnya komunikasi serial merupakan komunikasi dimana pengiriman data dilakukan per bit sehingga lebih lambat dibandingkan komunikasi paralel, atau dengan kata lain komunikasi serial merupakan salah satu metode komunikasi data di mana hanya satu bit data yang dikirimkan melalui seuntai kabel pada suatu waktu tertentu. Pada dasarnya komunikasi serial adalah kasus khusus komunikasi paralel dengan nilai $n = 1$, atau dengan kata lain adalah suatu bentuk komunikasi paralel dengan jumlah kabel hanya satu dan hanya mengirimkan satu bit data secara simultan. Hal ini dapat disandingkan dengan komunikasi paralel yang sesungguhnya di mana n -bit data dikirimkan bersamaan, dengan nilai umumnya $8 \leq n \leq 128$.

Komunikasi serial ada dua macam, asynchronous serial dan synchronous serial. Synchronous serial adalah komunikasi dimana hanya ada satu pihak (pengirim atau penerima) yang menghasilkan clock dan mengirimkan clock tersebut bersama-sama dengan data. Contoh penggunaan synchronous serial terdapat pada transmisi data keyboard. Asynchronous serial adalah komunikasi dimana kedua pihak (pengirim dan penerima) masing-masing menghasilkan clock namun hanya data yang ditransmisikan, tanpa clock. Agar data yang dikirim sama dengan data yang diterima, maka kedua frekuensi clock harus sama dan harus terdapat sinkronisasi. Setelah adanya sinkronisasi, pengirim akan mengirimkan datanya sesuai dengan frekuensi clock pengirim dan penerima akan membaca data sesuai dengan frekuensi clock penerima. Contoh penggunaan asynchronous serial adalah pada Universal Asynchronous Receiver Transmitter (UART) yang digunakan pada serial port (COM) komputer.

Antarmuka Kanal serial lebih kompleks atau sulit dibandingkan dengan antarmuka melalui kanal paralel, hal ini disebabkan karena:

- a. Dari Segi perangkat keras: adanya proses konversi data paralel menjadi serial atau sebaliknya menggunakan piranti tambahan yang disebut UART (Universal Asynchronous Receiver/Transmitter) dan
- b. Dari Segi perangkat lunak: lebih banyak register yang digunakan atau terlibat

Namun di sisi lain antarmuka kanal serial menawarkan beberapa kelebihan dibandingkan secara paralel, antara lain:

- a. Kabel untuk komunikasi serial bisa lebih panjang dibandingkan dengan paralel; data-data dalam komunikasi serial dikirimkan untuk logika '1' sebagai tegangan -3 s/d -25 volt dan untuk logika '0' sebagai tegangan +3 s/d +25 volt, dengan demikian tegangan dalam komunikasi serial memiliki ayunan tegangan maksimum 50 volt, sedangkan pada komunikasi paralel hanya 5 volt. Hal ini menyebabkan gangguan pada kabel-kabel panjang lebih mudah diatasi dibandingkan pada paralel.
- b. Jumlah kabel serial lebih sedikit; Anda bisa menghubungkan dua perangkat komputer yang berjauhan dengan hanya 3 kabel untuk

konfigurasi null modem, yaitu TXD (saluran kirim), RXD(saluran terima) dan Ground, bayangkan jika digunakan teknik paralel akan terdapat 20 – 25 kabel. Namun pada masing-masing komputer dengan komunikasi serial harus dibayar “biaya” antarmuka serial yang agak lebih mahal.

- c. Banyaknya piranti saat ini (palmtop, organizer, hand-phone dan lainlain) menggunakan teknologi infra merah untuk komunikasi data, dalam hal ini pengiriman datanya dilakukan secara serial. IrDA-1 (spesifikasi infra merah pertama) mampu mengirimkan data dengan laju 115,2 kbps dan Konsep Komunikasi Serial 2 dibantu dengan piranti UART, hanya panjang pulsa berkurang menjadi 3/16 dari standar RS-232 untuk menghemat daya.
- d. Untuk teknologi embedded system, banyak mikrokontroler yang dilengkapi dengan komunikasi serial (baik seri RISC maupun CISC) atau Serial Communication Interface (SCI); dengan adanya SCI yang terpadu pada IC mikrokontroler akan mengurangi jumlah pin keluaran, sehingga hanya dibutuhkan 2 pin utama TxD dan RxD (di luar acuan ground).

2.6.1 Komunikasi Serial Arduino

Komunikasi serial Arduino adalah Komunikasi antara Arduino Uno dan Komputer dapat dilakukan melalui port serial (via USB). Dalam hal ini, Arduino Uno tidak hanya bisa membaca data dari komputer yang ada di port serial, melainkan juga dapat mengirim data ke komputer. Sehingga komunikasi yang dilakukan bersifat dua arah.

Pada Arduino IDE menyesuaikan fasilitas untuk melakukan komunikasi dua arah tersebut melalui serial monitor. Dengan menggunakan fasilitas ini, dapat dikirimkan data ke Arduino Uno dan sebaliknya dapat membaca kiriman dari arduino uno. Tentu saja, hal ini memungkinkan dapat mengontrol Arduino Uno melalui komputer dan memantau sesuatu yang sedang terjadi di Arduino Uno. Sebagai contoh, saat mengirimkan isyarat untuk menghidupkan lampu atau memantau suhu yang terdeteksi oleh sensor suhu di Serial Monitor.

Jenis Perintah Komukasi serial Arduino

- a. `Serial.begin()` : berguna untuk menentukan kecepatan pengiriman dan penerimaan data melalui port serial. Kecepatan yang umum digunakan adalah 9600 bit per detik (9600 bps). Namun, kecepatan hingga 115.200 didukung oleh Arduino Uno. Contoh : `Serial.begin(9600)`.
- b. `Serial.end()` : digunakan untuk menghentikan komunikasi serial
- c. `Serial.available` : berguna untuk menghasilkan jumlah byte di port serial yang belum terbaca. Jika port serial dalam keadaan kosong, fungsi ini dapat menghasilkan nilai nol.
- d. `Serial.read()` : berguna untuk membaca satu byte data yang terdapat di port serial. Setelah pemanggilan `Serial.read()`, jumlah data di port serial berkurang satu. Untuk membaca seluruh data, diperlukan perintah seperti berikut :


```
int data;
while (Serial.available()){
  data = Serial.read();
}
```
- e. `Serial.print(data)` : berfungsi untuk mengirimkan data ke port serial. Apabila argumen format disertakan, data yang dikirim akan menyesuaikan dengan format tersebut. Dalam hal ini, format yang digunakan bisa berupa:
 - f. DEC : format desimal atau basis 10
 - g. HEX : format hexadesimal atau basis 16
 - h. OCT : format oktal atau basis 8
 - i. BIN : format biner atau basis 2
 - j. jika argument hanya satu, hasilnya dalam bentuk string. Contoh :
 - k. `Serial.print(65);` //hasil nilai tertampil 65
 - l. `Serial.print(65,DEC)` //hasil nilai tertampil 65
 - m. `Serial.print(65, HEX);` //hasil nilai tertampil 41
 - n. `Serial.print(65, OCT)` //hasil nilai tertampil 101
 - o. `Serial.print(65, BIN)` //hasil nilai tertampil 1000001
 - p. `Serial.flush()` : berguna untuk mengosongkan data pembacaan yang ditaruh pada buffer.

- q. `Serial.parseFloat()` : berguna untuk bilangan titik mengambang atau real.
- r. `Serial.println(data)` : memiliki fungsi yang hampir sama dengan `serial print`, yang memberi efek perpindahan baris berikutnya.
- s. `Serial.parseInt()` : untuk menghasilkan nilai bulat.

2.7 Bahasa Pemrograman C

Bahasa C merupakan akar dari bahasa BCPL yang dikembangkan oleh Martin Richards pada tahun 1967. Kemudian berdasar pada bahasa BCPL ini Ken Thompson yang bekerja di *Bell Telephone Laboratories (Bell Labs)* mengembangkan bahasa B pada tahun 1970. Saat itu bahasa B telah berhasil diimplementasikan di komputer DEC PDP-7 dengan operating system (OS) UNIX. Pada tahun 1972, peneliti lain di *Bell Labs* bernama Dennis Ritchie menyempurnakannya menjadi bahasa C.

Pada tahun 1978, Dennis Ritchie bersama dengan Brian Kernighan mempublikasikan buku yang kemudian menjadi legenda dalam sejarah perkembangan bahasa C, yang berjudul *The C Programming Language*. Buku ini diterbitkan oleh Prentice Hall, dan pada saat ini telah diterjemahkan dalam berbagai bahasa di dunia. Boleh dikatakan bahwa buku ini adalah buku yang paling banyak direfer orang dan dijadikan buku panduan tentang pemrograman bahasa C sampai saat ini. Teknik dan gaya penulisan bahasa C yang merefer kepada buku ini kemudian terkenal dengan sebutan K&R C atau Classic C atau Common C.

2.7.1 Struktur Bahasa Pemrograman C Arduino

Berikut ini struktur minimal C Arduino :

```
//setup digunakan untuk inialisasi variable, mode pin dll
void setup()
// semua kode yang disini akan dibaca sekali oleh Arduino
{
statement
}
```

```

void loop()
//semua kode yang ada disini akan dibaca berulang kali (terus menerus)
oleh Arduino
{
Statement
}

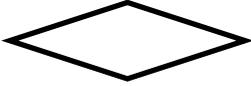
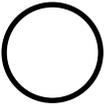
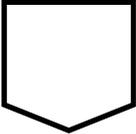
```

2.8 *Flowchart*

Flowchart (Diagram alir) Merupakan sebuah diagram dengan simbol-simbol grafis yang menyatakan aliran algoritma atau proses yang menampilkan langkah-langkah yang disimbolkan dalam bentuk kotak, beserta urutannya dengan menghubungkan masing masing langkah tersebut menggunakan tanda panah. Diagram ini bisa memberi solusi selangkah demi selangkah untuk penyelesaian masalah yang ada di dalam proses atau algoritma tersebut. *Flowchart* juga diartikan sebagai sebuah bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* disebut juga sebagai cara penyajian dari suatu algoritma. Notasi di *Flowchart* dapat dilihat pada Tabel 2.3 .

Tabel 2.3 Simbol-Simbol *Flowchart*

| Simbol | Keterangan | Fungsi |
|---|-------------|--|
|  | Terminal | Digunakan untuk menunjukkan awal dan akhir dari program. |
|  | <i>Flow</i> | Digunakan untuk arah aliran program. |
|  | Proses | Digunakan untuk mendefinisikan mekanisme perekam, proses, dan pelaporan. |

| | | |
|---|---|--|
|  | <p>Pilihan</p> | <p>Digunakan untuk mendefinisikan adanya 2 pilihan</p> |
|  | <p><i>Data Input-Output</i></p> | <p>Digunakan untuk mendefinisikan data yang dimasukkan dan data yang dikeluarkan ke dalam sistem</p> |
|  | <p><i>On Page Connector</i></p> | <p>Digunakan untuk menghubungkan alir data yang satu dengan yang lain dalam satu halaman</p> |
|  | <p><i>Off Page Connector</i></p> | <p>Digunakan sebagai penghubung bagian-bagian <i>flowchart</i> pada halaman yang berbeda</p> |
|  | <p>Predefined Process (Sub Program)</p> | <p>Permulaan sub program / proses menjalankan sub program</p> |
|  | <p>Preparation</p> | <p>Proses inisialisasi / pemberian harga awal</p> |

| | | |
|---|---------------------------|--|
|  | Documents Symbol | Sebuah dokumen dicetak atau laporan. |
|  | Multiple Documents Symbol | Merupakan beberapa dokumen dalam proses. |