

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Robot

Robot Berasal dari kata “*robot*” yang dalam bahasa Ceko berarti budak, pekerja atau kuli. Pertama kali kata “*robot*” diperkenalkan oleh Karl Capek dalam sebuah pentas sandiwara pada tahun 1921 yang berjudul RUR (*Rossum’s Universal Robot*). Pentas ini mengisahkan mesin yang menyerupai manusia yang dapat bekerja tanpa lelah yang kemudian memberontak dan menguasai manusia. Istilah “robot” ini kemudian mulai terkenal dan digunakan untuk menggantikan istilah yang dikenal saat itu, yaitu *automation*. (Adjie, 2009)

Sedangkan robot menurut *Webster*, robot adalah suatu peralatan otomatis yang dapat menirukan gerakan manusia. Sedangkan istilah Robotik berdasarkan *Webster* adalah : "Teknologi yang berhubungan dengan mendesain, membuat, dan mengoperasikan robot." (Pitowarno Endro, 2006)

Robot merupakan suatu hasil dari kemajuan teknologi yang dapat berbentuk macam – macam misalnya robot berbentuk kendaraan, hewan, bahkan berbentuk manusia. Robot ini pada umumnya diciptakan untuk dapat mempermudah pekerjaan manusia. (Adjie, 2009)

Unsur utama dalam kata “robot” adalah :

1. Seperangkat peralatan/*device*/mesin.
2. Dapat diprogram.
3. Bekerja/bergerak secara otomatis.
4. Mampu melaksanakan tugas tertentu sesuai program.

Tujuan pembuatan robot adalah untuk menggantikan tenaga manusia dan pekerjaan yang tidak dapat dilakukan manusia. Selain itu robot juga diprogram agar dapat bekerja secara otomatis. Keunggulan sistem robot dibandingkan dengan manusia :

1. Kuat
2. Tidak lelah

3. Dapat bekerja dengan tekanan tinggi
4. Tahan terhadap lingkungan berbahaya.

2.1.1 Struktur Robot

Pada umumnya sebuah robot memiliki struktur sebagai berikut :

1. Manipulator

- Badan (*body*)
- Pergelangan (*wrist*)
- Lengan (*arm*)

2. End Effector

Penjepit (*gripper*), bor, dll.

3. Penggerak (*Driver*)

Motor, hidrolik, *pneumatik*.

4. Sensor

Jarak, panas, kecepatan, posisi, dll.

5. Pengendali (*Controller*)

PC/komputer, mikrokontroler, mikroprosesor, dll.

2.1.2 Penggerak Robot

Macam – macam tenaga penggerak robot, yaitu :

1. Elektris

Tenaga penggeraknya menggunakan jenis motor DC, motor AC, motor *stepper* dan motor servo.

2. Hidrolis

Penggerak robot pada jenis ini menggunakan semacam cairan untuk menggerakkan. Tenaganya diperoleh dari pompa hidrolik dan pergerakannya diatur oleh *solenoid valve* yang kemudian diteruskan ke bagian aktuator melalui piston.

3. *Pneumatis*

Sama seperti penggerak hidrolis tetapi medium yang dipakai untuk menyalurkan tenaga berupa gas/udara.

2.2 Bahasa Pemograman

Software adalah suatu komponen di dalam suatu sistem data yang berupa program-program atau instruksi untuk mengontrol suatu sistem. Pada umumnya istilah *software* menyatakan cara-cara yang menghasilkan hubungan yang lebih efisien antara manusia dan komputer. (Rusmawan, 2004:65)

Dapat disimpulkan bahwa *software* merupakan rangkaian instruksi program yang diperlukan untuk menghasilkan informasi yang ditulis dengan bahasa khusus yang dimengerti oleh komputer dan manusia. Dalam melakukan pemograman banyak *software* yang dapat digunakan, salah satu perancangan robot manual ini menggunakan bahasa pemograman *software* arduino

2.2.1 Pengenalan Arduino

Arduino didefinisikan sebagai sebuah *platform* elektronik yang *open source*, berbasis pada *software* dan *hardware* yang fleksibel dan mudah digunakan, yang ditujukan untuk seniman, desainer, *hobbies* dan setiap orang yang tertarik dalam membuat objek atau lingkungan yang interaktif (Artanto, 2012:1)

Arduino sebagai sebuah *platform* komputasi fisik (*Physical Computing*) yang *open source* pada *board input output* sederhana, yang dimaksud dengan *platform* komputasi fisik disini adalah sebuah sistem fisik yang interaktif dengan penggunaan *software* dan *hardware* yang dapat mendeteksi dan merespon situasi dan kondisi.

Menurut Artanto (2012:2), kelebihan arduino dari *platform hardware* mikrokontroler lain adalah:

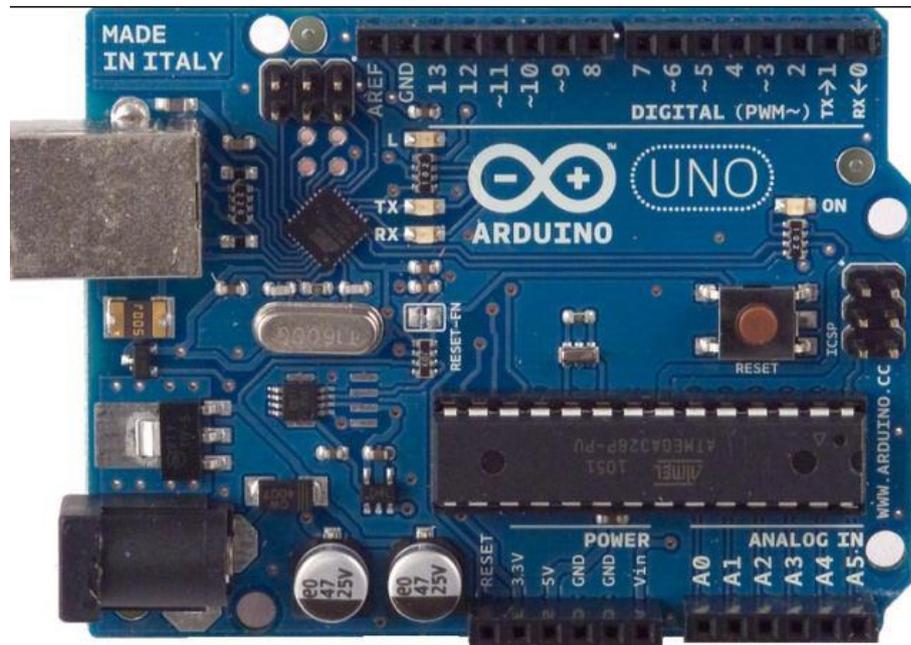
1. IDE Arduino merupakan *multiplatform*, yang dapat dijalankan di berbagai sistem operasi, seperti *Windows*, *Macintosh* dan *Linux*.
2. IDE Arduino dibuat berdasarkan pada IDE *Processing*, yang sederhana sehingga mudah digunakan.

3. Pemrograman arduino menggunakan kabel yang terhubung dengan *port* USB, bukan *port* serial. Fitur ini berguna karena banyak komputer yang sekarang ini tidak memiliki *port* serial.
4. Arduino adalah *hardware* dan *software open source* pembaca bisa mendownload *software* dan gambar rangkaian arduino tanpa harus membayar ke pembuat arduino.
5. Biaya *hardware* cukup murah, sehingga tidak terlalu menakutkan untuk membuat kesalahan.
6. Proyek arduino ini dikembangkan dalam lingkungan pendidikan sehingga bagi pemula akan lebih cepat dan mudah mempelajarinya.
7. Memiliki begitu banyak pengguna dan komunitas di internet dapat membantu setiap kesulitan yang dihadapi.

2.2.2 Sejarah Arduino

Proyek Arduino dimulai pertama kali di Ovre, Italy pada tahun 2005. Tujuan proyek ini awalnya untuk membuat peralatan *control* interaktif dan modul pembelajaran bagi siswa yang lebih murah dibandingkan dengan *prototype* yang lain. Pada tahun 2010 telah terjual dari 120 unit Arduino. Arduino yang berbasis *open source* melibatkan tim pengembang. Pendiri arduino itu Massimo Banzi dan David Cuartielles, awalnya mereka memberi nama proyek itu dengan sebutan arduino dari ivrea tetapi seturut perkembangan zaman nama proyek itu diubah menjadi Arduino.

Arduino dikembangkan dari thesis hernando Barragan di desain interaksi institute Ivrea. Arduino dapat menerima masukan dari berbagai macam sensor dan juga dapat mengontrol lampu, motor dan aktuator lainnya. Mikrokontroler pada *board* arduino di program dengan menggunakan bahasa pemrograman arduino (*based on wiring*) dan IDE arduino (*based on processing*). Proyek arduino dapat berjalan sendiri atau juga bisa berkomunikasi dengan *software* yang berjalan pada komputer.



Gambar 2.1 Arduino

(Sumber: <http://www.tobuku.com/docs/Arduino-Pengenalan.pdf>)

Papan Arduino merupakan papan mikrokontroler yang berukuran kecil atau dapat diartikan juga dengan suatu rangkaian berukuran kecil yang didalamnya terdapat komputer berbentuk suatu chip yang kecil. Pada Gambar 2.1 dapat dilihat sebuah papan Arduino dengan beberapa bagian komponen didalamnya.

Pada *hardware* arduino terdiri dari 20 pin yang meliputi:

a. 14 pin IO Digital (pin 0–13)

Sejumlah pin digital dengan nomor 0–13 yang dapat dijadikan input atau output yang diatur dengan cara membuat program IDE.

b. 6 pin Input Analog (pin 0–5)

Sejumlah pin analog bernomor 0–5 yang dapat digunakan untuk membaca nilai input yang memiliki nilai analog dan mengubahnya ke dalam angka antara 0 dan 1023.

c. 6 pin Output Analog (pin 3, 5, 6, 9, 10 dan 11)

Sejumlah pin yang sebenarnya merupakan pin digital tetapi sejumlah pin tersebut dapat diprogram kembali menjadi pin output analog dengan cara membuat programnya pada IDE.

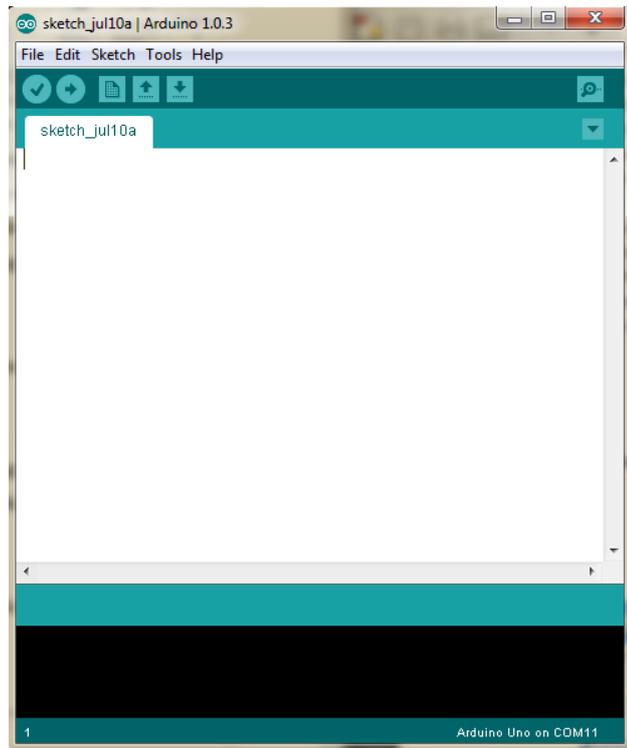
Papan Arduino Uno dapat mengambil daya dari USB *port* pada komputer dengan menggunakan USB *charger* atau dapat pula mengambil daya dengan menggunakan suatu AC adapter dengan tegangan 9 volt. Jika tidak terdapat power supply yang melalui AC adapter, maka papan Arduino akan mengambil daya dari USB *port*. Tetapi apabila diberikan daya melalui AC adapter secara bersamaan dengan USB port maka papan Arduino akan mengambil daya melalui AC adapter secara otomatis.

2.2.3 *Software* Arduino

Software arduino yang digunakan adalah *driver* dan IDE, walaupun masih ada beberapa *software* lain yang sangat berguna selama pengembangan arduino. IDE atau *Integrated Development Environment* suatu program khusus untuk suatu komputer agar dapat membuat suatu rancangan atau sketsa program untuk papan Arduino. IDE arduino merupakan *software* yang sangat canggih ditulis dengan menggunakan java. IDE arduino terdiri dari:

1. Editor Program

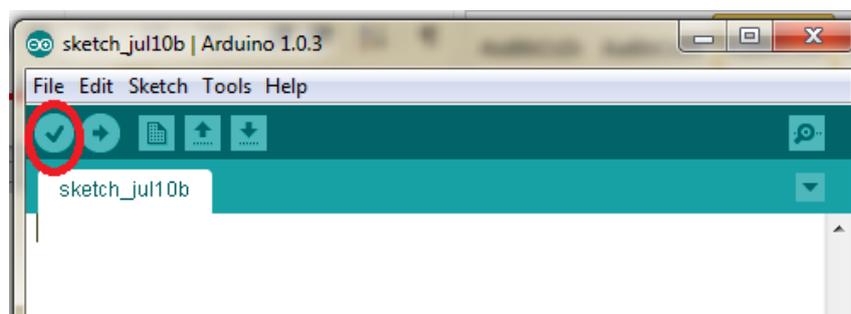
Sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *processing*.



Gambar 2.2 Tampilan Jendela Editor Program

2. Compiler

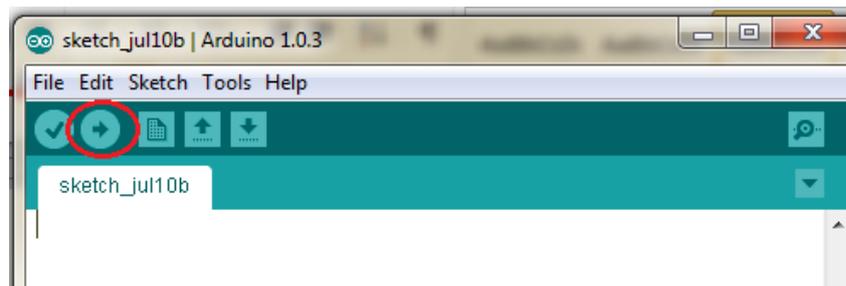
Sebuah modul yang mengubah kode program menjadi kode biner bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *processing*.



Gambar 2.3 Tampilan Jendela *Compiler*

3. Uploader

Sebuah modul yang memuat kode biner dari komputer ke dalam memory di dalam papan arduino.



Gambar 2.4 Tampilan Jendela *Uploader*

Dalam bahasa pemrograman arduino ada tiga bagian utama yaitu struktur, variabel dan fungsi (Artanto, 2012:27):

1. Struktur Program Arduino

Struktur dasar bahasa pemrograman arduino hanya terdiri dari dua bagian :

```

Void setup()
{
    // Statement;
}
Void loop()
{
    // Statement;
}

```

di eksekusi satu kali

di eksekusi terus menerus

a. Kerangka Program

Kerangka program arduino sangat sederhana, yaitu terdiri atas dua blok. Blok pertama adalah *void setup()* dan blok kedua adalah *void loop()*.

1). Blok *void setup ()*

Berisi kode program yang hanya dijalankan satu kali ketika program pertama kali di jalankan. Ini digunakan untuk mendefinisikan mode pin atau memulai komunikasi serial. Fungsi `setup()` harus di ikut sertakan dalam program walaupun tidak ada statement yang di jalankan.

```
void setup()
{
  Serial.begin(9600);
  pinMode (M1in1,OUTPUT);
  pinMode (M1in2,OUTPUT);
  pinMode (pwm1,OUTPUT);
  pinMode (pwm2,OUTPUT);
  pinMode (M2in1,OUTPUT);
  pinMode (M2in2,OUTPUT);
}
```

- `pinMode()` = berfungsi untuk mengatur fungsi sebuah pin sebagai INPUT atau OUTPUT.
- `Serial.begin(9600)` = digunakan untuk mengaktifkan fitur UART dan menginisialisasinya.

2). Blok *void loop()*

Setelah melakukan fungsi `setup()` maka secara langsung akan melakukan fungsi `loop()` secara berurutan dan melakukan instruksiinstruksi yang ada dalam fungsi `loop()`.

```
void loop ()
{
  digitalWrite(M1in1,LOW);
  digitalWrite(M1in2,LOW);
  digitalWrite(M2in1,LOW);
  digitalWrite(M2in2,LOW);
}
```

```

    maju();
    delay(2000);
    mundur();
    delay(2000);
    belokkanan();
    delay(2000);
    belokkiri();
    delay(2000);
}

```

- `digitalWrite()` = Berfungsi untuk memberikan nilai LOW atau HIGH pada sebuah pin *OUTPUT*
- Fungsi `delay` = Berfungsi untuk memberikan jeda dalam satuan milidetik

b. Sintaks Program

Baik blok `void setup loop ()` maupun blok `function` harus diberi tanda kurung kurawal buka “{“ sebagai tanda awal program di blok itu dan kurung kurawal tutup “}” sebagai tanda akhir program.

2. Variabel

Sebuah program secara garis besar dapat didefinisikan sebagai instruksi untuk memindahkan angka dengan cara yang cerdas dengan menggunakan sebuah variabel.

3. Fungsi

Pada bagian ini meliputi fungsi input *output digital*, *input output analog*, *advanced I/O*, fungsi waktu, fungsi matematika serta fungsi komunikasi.

Pada proses *Uploader* dimana pada proses ini mengubah bahasa pemrograman yang nantinya dcompile oleh avr-gcc (*avr-gcc compiler*) yang hasilnya akan disimpan kedalam papan arduino.

Avr-gcc compiler merupakan suatu bagian penting untuk *software* bersifat *open source*. Dengan adanya *avr-gcc compiler*, maka akan membuat bahasa pemrograman dapat dimengerti oleh mikrokontroler. Proses terakhir ini sangat penting, karena dengan adanya proses ini maka akan membuat proses pemrograman mikrokontroler menjadi sangat mudah.

Berikut ini merupakan gambaran siklus yang terjadi dalam melakukan pemrograman Arduino:

1. Koneksikan papan Arduino dengan komputer melalui *USB port*.
2. Tuliskan sketsa rancangan suatu program yang akan dimasukkan ke dalam papan Arduino.
3. Upload sketsa program ke dalam papan Arduino melalui kabel USB dan kemudian tunggu beberapa saat untuk melakukan restart pada papan Arduino.
4. Papan Arduino akan mengeksekusi rancangan sketsa program yang telah dibuat dan di-upload ke papan Arduino.

2.3 Komunikasi Data

Pada saat ini kegiatan pemrosesan data sudah semakin luas, baik yang berorientasi kepada ilmu pengetahuan, komersil/bisnis maupun kegiatan pemerintahan, sehingga data yang diolahpun akan bermacam-macam sesuai dengan bidang pekerjaan tersebut.

Dari keterangan diatas dapat diambil kesimpulan bahwa data tersebut merupakan bahan yang akan diolah menjadi suatu bentuk yang lebih berguna dan lebih mempunyai arti. Sedangkan informasi adalah hasil pengolahan data atau hasil proses dari data tersebut. Proses perubahan dari data menjadi informasi merupakan fungsi utama dari pengolahan data. Cara pengolahan data menjadi

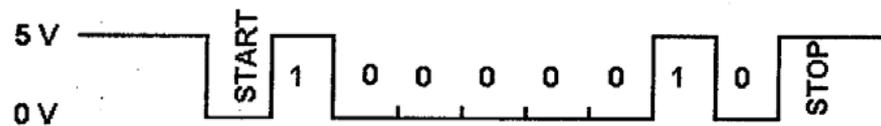
informasi tersebut bisa bermacam-macam misalnya secara manual (sempoa), mekanis (*register*), elektris (kalkulator) dan elektronik (komputer).

Sedangkan pengertian komunikasi data sendiri adalah transmisi data elektronik melalui beberapa media. Media tersebut dapat berupa kabel koaksial, fiber optik dan sebagainya. Sistem yang memungkinkan terjadinya transmisi data seringkali disebut jaringan komunikasi data.

2.3.1 Komunikasi Data Serial

Dikenal dua cara komunikasi data secara serial, yaitu komunikasi data serial secara sinkron dan komunikasi data serial secara asinkron. Pada komunikasi data serial sinkron, *clock* dikirimkan bersama-sama dengan data serial, sedangkan komunikasi data serial asinkron, *clock* tidak dikirimkan bersama data serial, tetapi dibangkitkan secara sendiri-sendiri baik pada sisi pengirim (*transmitter*) maupun pada sisi penerima (*receiver*). Komunikasi data serial ini dikerjakan oleh UART (*Universal Asynchronous Receiver/Transmitter*). IC UART dibuat khusus untuk mengubah data paralel menjadi data serial dan menerima data serial yang kemudian diubah kembali menjadi data paralel.

Pada UART, kecepatan pengiriman data (*baud rate*) dan *fase clock* pada sisi *transmitter* dan pada sisi *receiver* harus sinkron. Untuk itu diperlukan sinkronisasi antara *transmitter* dan *receiver*. Hal ini dilakukan oleh bit 'Start' dan bit 'Stop'. Ketika saluran transmisi dalam keadaan *idle*, *output* UART adalah dalam keadaan logika '1'. Ketika *transmitter* ingin mengirimkan data, *output* UART akan diset lebih dulu ke logika '0' untuk waktu satu bit. Sinyal ini pada *receiver* akan dikenali sebagai sinyal 'Start' yang digunakan untuk mensinkronkan fase clocknya sehingga sinkron dengan *fase clock transmitter*. Selanjutnya, data akan dikirimkan secara serial dari bit paling rendah (bit 0) sampai bit tertinggi. Selanjutnya akan dikirim sinyal 'Stop' sebagai akhir dari pengiriman data serial. Cara pemberian kode data yang disalurkan tidak ditetapkan secara pasti. Berikut adalah contoh pengiriman huruf 'A' dalam format ASCII (41 heksa / 1000001 biner) tanpa bit paritas.



Gambar 2.5 Pengiriman huruf 'A' tanpa bit paritas

(Sumber: *Makalah-Komunikasi-Data-Serial-Dan-Paralel.pdf*)

Kecepatan transmisi (*baud rate*) dapat dipilih bebas dalam rentang tertentu. *Baud rate* yang umum dipakai adalah 110, 135, 150, 300, 600, 1200, 2400, dan 9600 (bit/detik). Dalam komunikasi data serial, *baud rate* dari kedua alat yang berhubungan harus diatur pada kecepatan yang sama. Selanjutnya, harus ditentukan panjang data (6,7, atau 8 bit), paritas (genap, ganjil, atau tanpa baris), dan jumlah bit 'Stop' (1, 1 ½, atau 2 bit).

2.3.2 Komunikasi Data Pararel

Port paralel atau port printer sebenarnya terdiri dari tiga bagian yang masing-masing diberi nama sesuai dengan tugasnya dalam melaksanakan pencetakan pada printer. Tiga bagian tersebut adalah *Data Port* (DP), *Printer Control* (PC) dan *Printer Status* (PS). DP digunakan untuk mengirim data yang harus dicetak oleh printer, PC digunakan untuk mengirimkan kode-kode kontrol dari komputer ke printer, misalnya kode kontrol untuk menggulung kertas, dan PS digunakan untuk mengirimkan kode-kode status printer ke komputer, misalnya untuk menginformasikan bahwa kertas telah habis.

DP, PC, dan PS sebenarnya adalah port-port 8 bit, namun hanya DP yang benar-benar 8 bit. Untuk PC dan PS, hanya beberapa bit saja yang dipakai yang berarti hanya beberapa bit saja dari port-port ini yang dapat kita manfaatkan untuk keperluan *interfacing*. Port PC adalah port baca/tulis (*read/write*), PS adalah port baca saja (*read only*), sedangkan port DP merupakan port baca/tulis juga. Akan tetapi kemampuan ini hanya dimiliki oleh *Enhanced Parallel Port* (EPP), sedangkan port paralel standar hanya memiliki kemampuan tulis saja. Pada EPP, pengaturan arah jalur data DP dilakukan lewat bit 5 PC. Jika bit 5 PC bernilai 0,

maka jalur data 2 arah DP menjadi output dari port paralel, sebaliknya jika bit 5 PC bernilai 1, maka jalur data 2 arah DP menjadi input port paralel.

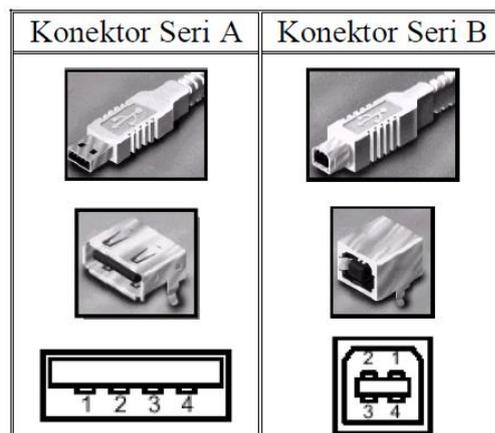
2.4 Universal Serial Bus (USB)

USB merupakan *port* masukan/keluaran baru yang dibuat untuk mengatasi kekurangan-kekurangan *port* serial maupun paralel yang sudah ada. USB dibuat dengan kelebihan-kelebihan sebagai berikut (John Hyde, 1999) :

1. *Hot-plugable*, yang berarti piranti masukan/keluaran yang menggunakan USB dapat ditambahkan ketika PC menyala.
2. Mudah digunakan karena piranti masukan/keluaran yang terpasang dikenali oleh PC menggunakan *driver* yang sesuai kemudian konfigurasinya akan dikerjakan secara otomatis.
3. Semua piranti dipasang menggunakan satu tipe konektor.
4. Kecepatan USB sangat tinggi, dapat mencapai 12 Mbps yang tentunya jauh lebih cepat dibanding port serial dan paralel yang ada saat ini.
5. Jumlah piranti yang dapat dipasang pada 1 PC mencapai 127 piranti (dengan bantuan hub yang dapat dipasang sampai 5 tingkat), suatu batasan yang sangat tinggi untuk ekspansi piranti masukan/keluaran.
6. Piranti dengan USB dapat menggunakan catu dari PC (untuk penggunaan arus tidak lebih dari 500mA) sehingga tidak membutuhkan tambahan catu daya luar.
7. Hemat listrik karena piranti dapat mati secara otomatis apabila tidak digunakan (PC dalam keadaan *suspend*).
8. Adanya deteksi dan pemulihan kesalahan yang handal. Kesalahan data dideteksi dan transaksi diulang lagi untuk memastikan data dikirim/diterima dengan benar.
9. Merupakan piranti eksternal PC sehingga tidak perlu membuka kotak PC atau merancang suatu kartu antarmuka dalam penggunaan piranti masukan/keluaran dengan USB.

Komunikasi pada USB dilakukan secara serial. Serial lebih dipilih dibanding dengan paralel karena kebutuhan kabel yang lebih sedikit, sehingga lebih murah, dan lebih mudah diterapkan dalam konfigurasi dinamik. Yang dimaksud dengan konfigurasi dinamik adalah suatu sistem masukan/keluaran dapat dipasang atau dikonfigurasi ulang dengan memasang atau melepas kabel ketika PC bekerja. Pada konfigurasi dinamik tidak diperlukan booting ulang. (John Hyde, 1999)

Hubungan piranti masukan/keluaran dengan PC melalui USB dilakukan secara asimetrik yang berarti konektor pada kedua ujung kabel tidak sama sehingga harus diketahui ujung mana yang dipasang pada master dan ujung mana yang dipasang pada *slave*. Terminologi yang diadopsi oleh spesifikasi USB adalah “*upstream*” (menuju PC) dan “*downstream*” (menuju piranti masukan/keluaran). Ujung “*upstream*” mengatur protokol dan menginstruksikan ujung “*downstream*” untuk membalas pada waktu yang ditentukan. Konektor tipe A dipasang pada PC sedangkan tipe B dipasang pada piranti masukan/keluaran. Konektor pada USB memiliki 4 kaki, yaitu VCC +5V (atau sering disebut VBUS), Data- (D-), Data+ (D+), dan GND.



Gambar 2.6 Tipe Konektor USB

(Sumber: <http://eprints.undip.ac.id/25543/1/ML2F000599.pdf>)

Tabel 2.1 Susunan Kaki pada Konektor USB

| Nomor | Kaki |
|-------|------|
| 1 | VCC |
| 2 | D- |
| 3 | D+ |
| 4 | GND |

2.5 Mikrokontroler AVR Atmega8

Mikrokontroler merupakan sebuah sistem komputer yang seluruh atau sebagian besar elemennya dikemas dalam satu *chip* IC, sehingga sering disebut *single chip microcomputer*. Lebih lanjut, mikrokontroler merupakan sistem komputer yang mempunyai satu atau beberapa tugas yang sangat spesifik, berbeda dengan PC (*Personal Computer*) yang memiliki beragam fungsi. Perbedaan lainnya adalah perbandingan RAM dan ROM yang sangat berbeda antara komputer dengan mikrokontroler.

Mikrokontroler adalah sebuah *system microprocessor* dimana didalamnya sudah terdapat CPU, ROM, RAM, I/O, Clock dan peralatan internal lainnya yang sudah saling terhubung dan terorganisasi (teralamat) dengan baik oleh pabrik pembuatnya dan dikemas dalam satu chip yang siap pakai. Sehingga kita tinggal memprogram isi ROM sesuai aturan penggunaan oleh pabrik yang membuatnya. (Winoto, 2008:3)

AVR merupakan salah satu jenis mikrokontroler yang di dalamnya terdapat berbagai macam fungsi. Perbedaannya pada mikro yang pada umumnya digunakan seperti MCS51 adalah pada AVR tidak perlu menggunakan *oscillator eksternal* karena di dalamnya sudah terdapat *internal oscillator*. Selain itu kelebihan dari AVR adalah memiliki *Power-On Reset*, yaitu tidak perlu ada tombol reset dari luar karena cukup hanya dengan mematikan *supply*, maka secara otomatis AVR akan melakukan reset.

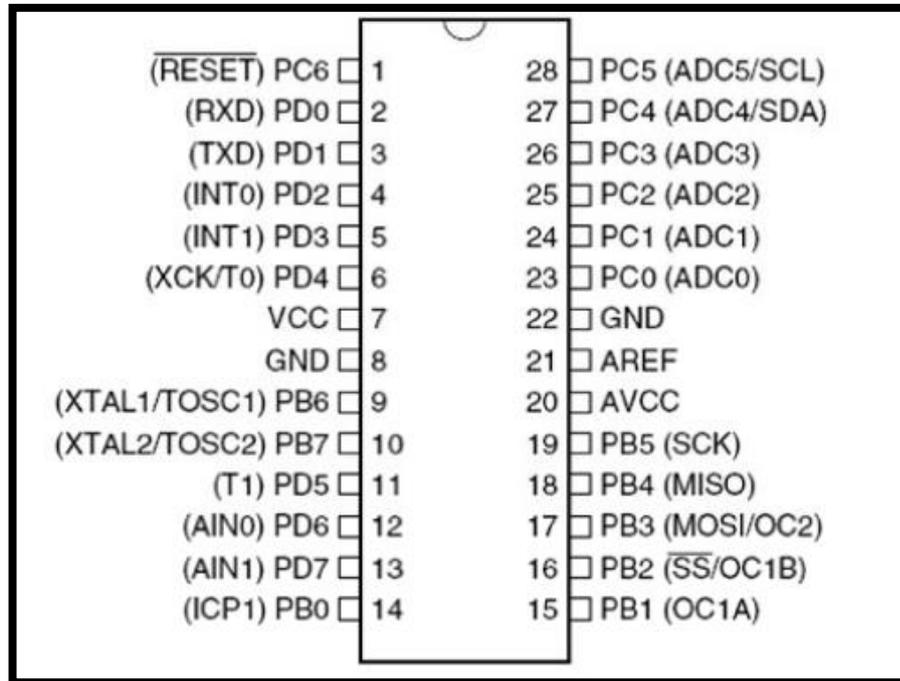
Untuk beberapa jenis AVR terdapat beberapa fungsi khusus seperti ADC, EEPROM sekitar 128 byte sampai dengan 512 *byte*. AVR ATmega8 adalah mikrokontroler CMOS 8-bit berarsitektur AVR RISC yang memiliki 8K *byte in-System Programmable Flash*. Mikrokontroler dengan konsumsi daya rendah ini mampu mengeksekusi instruksi dengan kecepatan maksimum 16MIPS pada frekuensi 16MHz. Jika dibandingkan dengan ATmega8L perbedaannya hanya terletak pada besarnya tegangan yang diperlukan untuk bekerja. Untuk ATmega8 tipe L, mikrokontroler ini dapat bekerja dengan tegangan antara 2,7 - 5,5 V sedangkan untuk ATmega8 hanya dapat bekerja pada tegangan antara 4,5 - 5,5 V.



Gambar 2.7 Atmega 8

(Sumber: <http://elib.unikom.ac.id/files/.pdf>)

2.5.1 Konfigurasi Pin Atmega8



Gambar 2.8 Konfigurasi Pin Atmega8

(Sumber : http://jbptunikompp-gdl-indrapurna-26711-5-unikom_i-i_5)

ATmega8 memiliki 28 Pin, yang masing-masing pin nya memiliki fungsi yang berbeda-beda baik sebagai port maupun fungsi yang lainnya. Berikut akan dijelaskan fungsi dari masing-masing kaki ATmega8.

- 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 *bi-directional* I/O dengan internal pull-up resistor. Sebagai *input*, pin-pin yang

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 pulsa minimum, maka akan menghasilkan suatu kondisi reset meskipun *clock*-nya tidak bekerja.

– Port D (PD7...PD0)

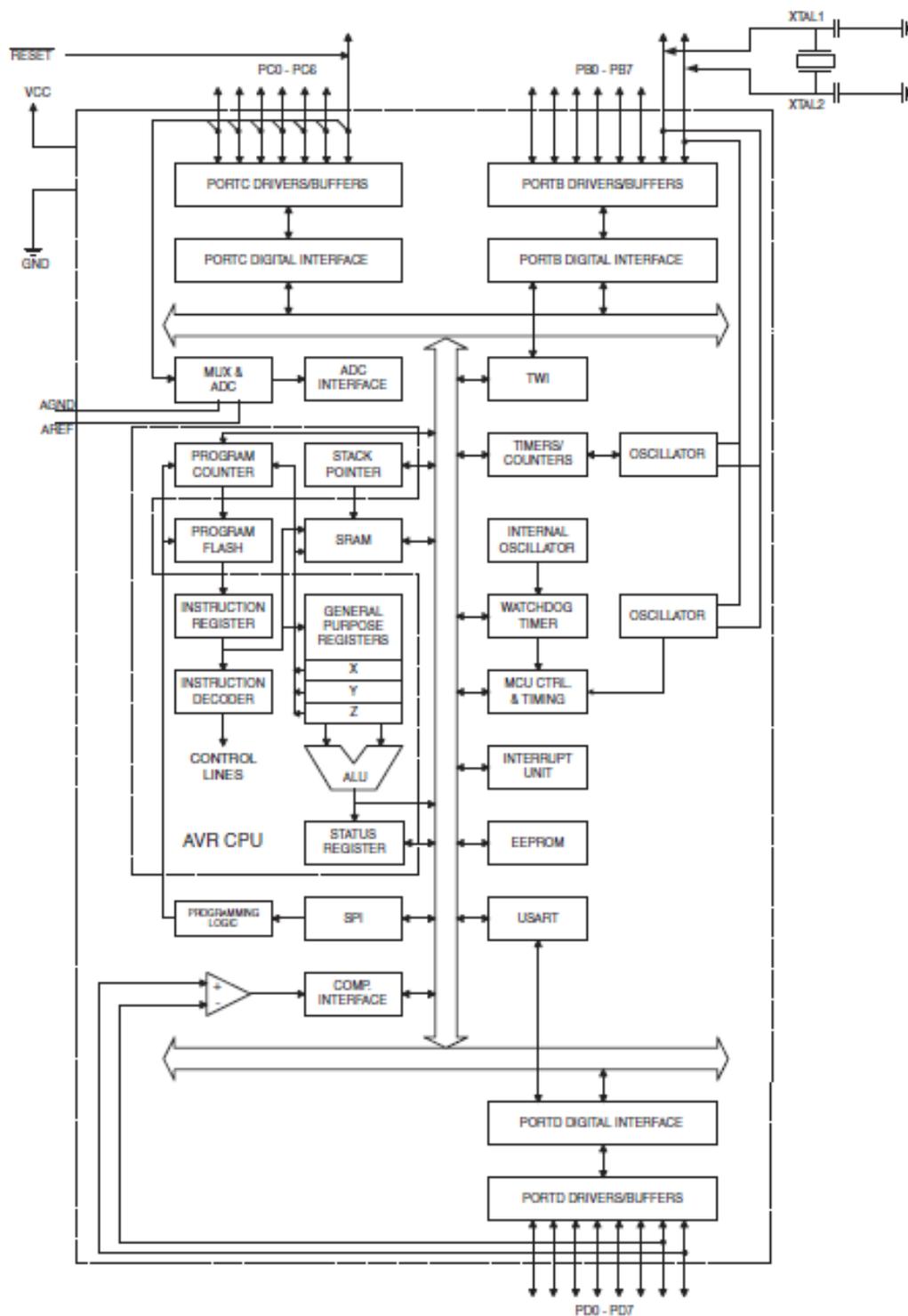
Port D merupakan *8-bit bi-directional I/O* dengan internal *pull-up* resistor. Fungsi dari port ini sama dengan port-port yang lain. Hanya saja pada port ini tidak terdapat kegunaan-kegunaan yang lain. Pada *port* ini hanya berfungsi sebagai masukan dan keluaran saja atau biasa disebut dengan I/O.

- 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.

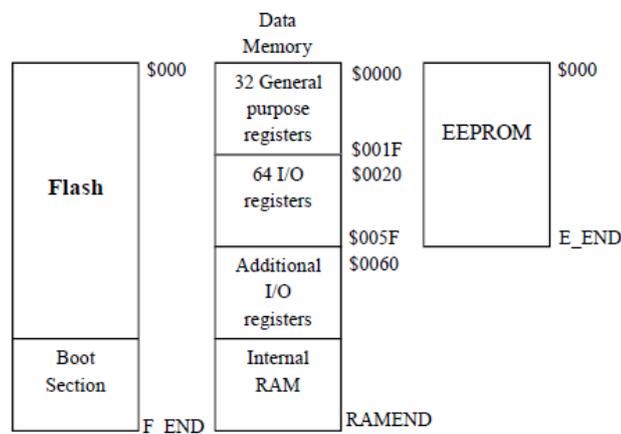


Gambar 2.9 Diagram Blok Mikrokontroler Atmega 8

(Sumber : http://jbptunikompp-gdl-indrapurna-26711-5-unikom_i-i_5)

Pada AVR status register mengandung beberapa informasi mengenai hasil dari kebanyakan hasil eksekusi instruksi aritmatik. Informasi ini digunakan untuk altering arus program sebagai kegunaan untuk meningkatkan performa pengoperasian. Register ini di-update setelah operasi ALU (*Arithmetic Logic Unit*) hal tersebut seperti yang tertulis dalam datasheet khususnya pada bagian *Instruction Set Reference*. Dalam hal ini untuk beberapa kasus dapat membuang penggunaan kebutuhan instruksi perbandingan yang telah didedikasikan serta dapat menghasilkan peningkatan dalam hal kecepatan dan kode yang lebih sederhana dan singkat. *Register* ini tidak secara otomatis tersimpan ketika memasuki sebuah rutin interupsi dan juga ketika menjalankan sebuah perintah setelah kembali dari interupsi. Namun hal tersebut harus dilakukan melalui *software*.

2.5.2 Memori AVR Atmega8



Gambar 2.10 Peta Memori Atmega 8

(Sumber: <http://elib.unikom.ac.id/files/.pdf>)

Memori atmega terbagi menjadi tiga yaitu :

1. Memori *Flash*

Memori flash adalah memori ROM tempat kode-kode program berada.

Kata flash menunjukkan jenis ROM yang dapat ditulis dan dihapus

secara elektrik. Memori flash terbagi menjadi dua bagian yaitu bagian aplikasi dan bagian *boot*. Bagian aplikasi adalah bagian kode-kode program aplikasi berada. Bagian *boot* adalah bagian yang digunakan khusus untuk *booting* awal yang dapat diprogram untuk menulis bagian aplikasi tanpa melalui *programmer/downloader*.

2. Memori Data

Memori data adalah memori RAM yang digunakan untuk keperluan program. Memori data terbagi menjadi empat bagian yaitu :

32 GPR (*General Purpose Register*) adalah register khusus yang bertugas untuk membantu eksekusi program oleh ALU (*Arithmatic Logic Unit*), dalam instruksi assembler setiap instruksi harus melibatkan GPR. Dalam bahasa C biasanya digunakan untuk variabel global atau nilai balik fungsi dan nilainilai yang dapat memperingan kerja ALU. Dalam istilah processor komputer sehari-hari GPR dikenal sebagai "*chace memory*". I/O register dan Additional I/O register adalah *register* yang difungsikan khusus untuk mengendalikan berbagai pheripheral dalam mikrokontroler seperti *pin port, timer/counter, usart* dan lain-lain. Register ini dalam keluarga mikrokontrol MCS51 dikenal sebagai SFR(*Special Function Register*).

3. EEPROM

EEPROM adalah memori data yang dapat mengendap ketika chip mati (*off*), digunakan untuk keperluan penyimpanan data yang tahan terhadap gangguan catu daya.

2.6 *Downloder USBasp*

USBasp *programmer* adalah suatu perangkat yang digunakan untuk menuliskan program (mendownload) pada mikrokontroler ATMEL AVR. Dengan menggunakan perangkat ini kita bias mengisikan file hex yang telah di Compile

(di ciptakan) oleh beberapa software seperti Codevision AVR. Banyak kelebihan yang kita miliki dengan menggunakan perangkat ini ,diantaranya:

1. Kecepatan mengisi program yaitu 5 kBytes/s
2. Tidak menggunakan kontroler tambahan seperti SMD khusus
3. Fasilitas SCK digunakan untuk mendownload target dengan kecepatan rendah (kurang dari 1,5MHz).
4. Tidak memerlukan supply tambahan dan dapat digunakan untuk mensupply power dari mikrokontroller.
5. Bisa digunakan untuk berbagai macam platform seperti Linux, Mac OS, Windows XP, Windows Vista dan Windows 7.

2.7 Flowchart

Flowchart (bagan alir) merupakan suatu bagan yang menggambarkan arus logika dari data yang akan diproses dari awal sampai akhir. Tujuan utama dari penggunaan *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurut rapid an jelas dengan menggunakan symbol-simbol yang standar. Tahap masalah yang disajikan harus jelas, sederhana, efektif dan tepat. Dalam penulisan *flowchart* dikenal dua metode, yaitu sistem *flowchart* dan program *flowchart*. (Tosin, 1994:16-25).

2.7.1 Sistem Flowchart

Sistem *Flowchart* merupakan diagram alir yang menggambarkan suatu sistem peralatan computer yang digunakan dalam proses pengolahan data serta hubungan antar peralatan tersebut. Sistem *flowchart* ini tidak digunakan untuk menggambarkan urutan langkah untuk memecahkan masalah, tetapi hanya untuk menggambarkan prosedur dalam system yang dibentuk.

Dalam menggambarkan *flowchart* biasanya digunakan symbol-simbol yang standar, tetapi pemograman juga dapat membuat symbol-simbol yang telah tersedia dirasa masih kurang. Dalam kasus ini pemograman harus melengkapi gambar *flowchart* tersebut dengan kamus symbol untuk menjelaskan arti dari

masing-masing symbol yang digunakan agar pemograman lain dapat mengetahui maksud dari symbol-simbol tersebut. (Tosin, 1994:16-25).

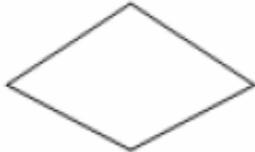
2.7.2 Program *Flowchart*

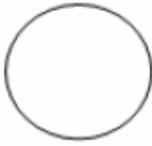
Program *Flowchart* merupakan diagram alir yang menggambarkan urutan logika dari suatu prosedur pemecahan masalah. Dalam menggambarkan program *flowchart*, telah tersedia simbol-simbol standar, tetapi seperti pada system *flowchart*, pemograman dapat menambah khasanah simbol-simbol tersebut, tetapi pemograman harus melengkapi penggambaran program *flowchart* dengan kamus symbol. (Tosin, 1994:16-25).

Perbedaan program *flowchart* dan sistem *flowchart* ialah pada sistem *flowchart* digunakan untuk menggambarkan urutan langkah untuk memecahkan masalah, tetapi hanya untuk menggambarkan prosedur dalam system yang dibentuk. Sedangkan system *flowchart* merupakan diagram alir yang menggambarkan urutan logika,

Berikut ini adalah gambar dari simbol-simbol yang standar yang digunakan pada program *flowchart*.

Tabel 2.2 Simbol-simbol yang digunakan pada *flowchart*
 (sumber: <http://www.smartdraw.com/specials/flowchart-symbols.htm>)

| | Simbol | Keterangan |
|--------------------|---|--|
| Terminal |  | Menyatakan dimana program dimulai dan mungkin juga berhenti. |
| Proses |  | Menyatakan beberapa manipulasi data. |
| Masukan / Keluaran |  | Menyatakan beberapa operasi masukan/keluaran data. |
| Keputusan |  | Menunjukkan perbandingan dua nilai. |
| Persiapan |  | Menyatakan beberapa manipulasi yang ditunjukkan dalam program sari dalam data. |
| Proses Terdefinisi |  | Menyatakan untuk proses yang detailnya dijelaskan terpisah. |

| | | |
|------------|--|--|
| Penghubung |  | Menyatakan sambungan ke atau dari bagian lain program yang terputus masih dalam halaman yang sama. |
| Garis Alur |  | Menyatakan urutan operasi. |
| |  | menambahkan keterangan yang menjelaskan langkah pemrosesan. |
| Penghubung |  | Menyatakan untuk menunjukkan hubungan arus proses yang terputus tetapi tidak pada halaman yang sama. |