

BAB II

LANDASAN TEORI

2.1. Sejarah Catur

Permainan di atas papan berisi 8 x 8 petak atau 64 petak ini berasal dari India sejak 500 Masehi, kemudian menyebar ke Persia dan masyarakat Arab. *Chess* atau catur menyebar ke Eropa ketika kekuasaan Islam pada awal abad pertengahan memasuki Eropa dari selatan Spanyol.

Bentuk buah catur sempat berubah. Awalnya bentuk buah catur mirip manusia, kini berubah menjadi abstrak. Ketika memasuki Eropa, buah catur kembali mengambil bentuk menyerupai manusia. Buah-buah catur mewakili sejumlah golongan pada abad pertengahan yaitu:

1. Buah Pion mewakili budak yang kala itu selalu mengorbankan jiwa dan raga.
2. Buah Benteng mewakili rumah dan tempat berlindung.
3. Buah Kuda mewakili ksatria yang senantiasa melindungi negara.
4. Buah Peluncur mewakili gereja yang menjadi lambang keagamaan di abad pertengahan.
5. Buah Ratu atau *Ster* mewakili Ratu yang merupakan wanita paling berkuasa pada masa itu.
6. Buah Raja mewakili Raja yang merupakan pucuk pimpinan dan menentukan kalah menang pertarungan.

Catur dimainkan dua orang. Masing-masing pemain memegang buah catur putih melawan buah catur hitam. Yang menjadi pemenang dalam catur adalah

pemain yang berhasil men-*skak* (membuat Raja tidak bisa melangkah kemana pun) atau mematikan Raja.

Masing-masing buah catur memiliki pola pergerakan yang berbeda. Pion hanya boleh berjalan satu kotak ke depan, kecuali langkah pertamanya, boleh dua kotak ke depan. Pion tidak boleh jalan mundur, namun Pion memakan musuhnya dengan langkah diagonal kiri atau kanan. Benteng berjalan lurus secara vertikal dan horizontal, sementara Menteri atau Peluncur berjalan maju mundur secara diagonal sesuai warna petak. Cara jalan Ratu merupakan kombinasi cara jalan Benteng dan Menteri. Kuda bisa melompati halangan di depannya, asalkan alur jalannya menyerupai huruf “L” sebanyak 4 kotak. Untuk Raja bisa berjalan ke segala arah sebanyak satu kotak (Magethi, 2009:12).

2.2. Ketentuan Permainan Catur

Permainan dilangsungkan di atas papan yang terdiri dari 8 kolom dan 8 baris kotak atau petak berwarna hitam dan putih (terang dan gelap) secara berselang seling. Permainan dimulai dengan 16 buah pada masing-masing pihak, yang disusun berbaris secara khusus pada masing-masing sisi papan catur secara berhadap-hadapan. Satu buah hanya bisa menempati satu petak. Pada bagian terdepan masing-masing barisan terdapat 8 Pion, diikuti di belakangnya dua Benteng, dua Kuda (dalam bahasa Inggris disebut *knight* atau ksatria), dua Menteri atau Peluncur (dalam bahasa Inggris disebut *bishop* atau uskup), Ratu atau *ster*, serta satu Raja (Magethi, 2009:25).

2.2.1. Pola Pergerakan Bidak Catur

Sebelum bertanding, pecatur memilih warna buah yang akan ia mainkan. Pemegang buah putih memulai langkah pertama, yang selanjutnya diikuti oleh pemegang buah hitam secara bergantian. Setiap langkah hanya boleh menggerakkan satu bidak saja (kecuali untuk Rokade di mana ada dua bidak yang digerakkan). Bidak dipindahkan ke petak kosong, ataupun yang ditempati oleh bidak lawan, yang berarti menangkapnya dan memindahkan bidak lawan dari permainan. Ada pengecualian, yaitu untuk gerakan *en passant*. Setiap bidak catur memiliki gerakan yang unik sebagai berikut (Magethi, 2009:32):

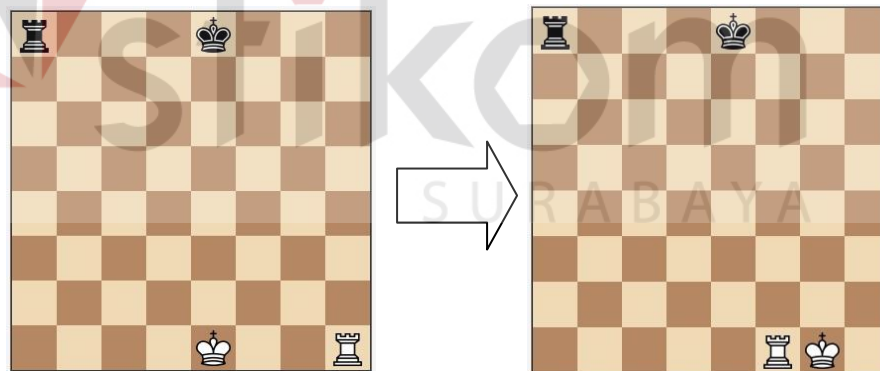
1. Raja dapat bergerak satu petak ke segala arah. Raja juga memiliki gerakan khusus yang disebut Rokade yang turut melibatkan sebuah Benteng.
2. Benteng dapat bergerak sepanjang petak horizontal maupun vertikal, tetapi tidak dapat melompati bidak lain. Seperti yang telah di atas, Benteng terlibat dalam gerakan Rokade.
3. Peluncur dapat bergerak sepanjang petak secara diagonal, tetapi tidak dapat melompati bidak lain.
4. Ratu memiliki gerakan kombinasi dari Benteng dan Peluncur.
5. Kuda memiliki gerakan mirip huruf L, yaitu memanjang dua petak dan melebar satu petak. Kuda adalah satu-satunya bidak yang dapat melompati bidak-bidak lain.
6. Pion dapat bergerak maju (arah lawan) satu petak ke petak yang tidak ditempati. Pada gerakan awal, Pion dapat bergerak maju dua petak. Pion juga dapat menangkap bidak lawan secara diagonal, apabila bidak lawan tersebut

berada satu petak di diagonal depannya. Pion memiliki dua gerakan khusus, yaitu gerakan menangkap *en passant* dan promosi.

2.2.2. Rokade

Rokade (dalam bahasa Inggris, *castling*) merupakan gerakan khusus dalam catur di mana Raja bergerak dua petak menuju Benteng di baris pertamanya, kemudian meletakkan Benteng pada petak terakhir yang dilalui Raja. Persyaratan Rokade adalah sebagai berikut:

1. Bidak Raja dan Benteng yang akan dilibatkan dalam Rokade harus belum pernah bergerak.
2. Tidak ada bidak lain di antara Raja dan Benteng.
3. Raja tidak sedang di-skak, dan petak-petak yang dilalui Raja tidak sedang diserang oleh bidak lawan. Berikut adalah gambar mengenai proses Rokade :

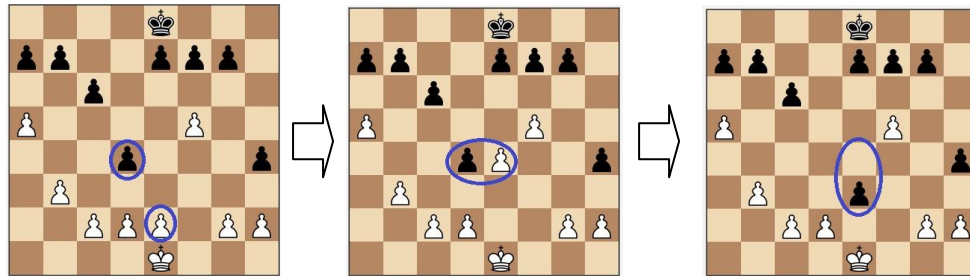


Gambar 2.1. Proses pergerakan Rokade
(Sumber: <http://www.chess.com/learn-how-to-play-chess.html>).

2.2.3. *En passant*

Ketika Pion bergerak dua petak maju dan ada Pion lawan yang berada satu petak dalam baris tujuan, maka Pion lawan dapat menangkap dan menempati petak yang baru saja dilalui Pion tersebut (seolah-olah Pion tersebut bergerak satu

petak maju). Namun demikian, gerakan ini hanya dapat dilakukan sesaat setelah gerakan Pion maju dua petak, atau hak lawan untuk melakukan gerakan *en passant* ini hilang. Berikut adalah contoh proses *en passant* berlangsung :

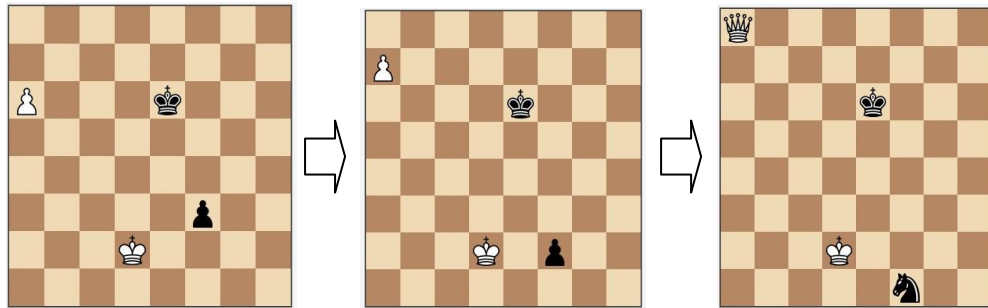


Gambar 2.2. Proses pergerakan *en passant*

(Sumber: <http://www.chess.com/learn-how-to-play-chess.html>).

2.2.4. Promosi

Ketika Pion telah maju hingga menempati baris paling akhir, berbarengan dengan gerakan maju tersebut, Pion dipromosikan dan harus ditukar dengan bidak berdasarkan keinginan pemain, yaitu Ratu, Benteng, Peluncur, ataupun Kuda dengan warna yang sama. Pada umumnya, Pion dipromosikan menjadi Ratu. Tidak ada peraturan yang membatasi bidak yang dipilih sebagai promosi, jadi dimungkinkan memiliki bidak yang melebihi jumlahnya waktu awal permainan (semisal, dua Ratu). Berikut adalah gambar dari contoh proses promosi berlangsung pada permainan catur :



Gambar 2.3. Proses pergerakan promosi
(Sumber: <http://www.chess.com/learn-how-to-play-chess.html>).

2.2.5. Skak

Ketika Raja sedang diserang oleh satu atau lebih bidak lawan, keadaan ini disebut dengan skak. Pemain yang Rajanya diskak harus menggerakkan Rajanya supaya tidak terserang. Hal ini dapat dilakukan dengan menangkap bidak lawan yang menyerang, menutup serangan lawan dengan menempatkan sebuah bidak di antaranya (apabila yang menyerang Ratu, Benteng, atau Peluncur dan ada petak kosong di antara Raja dan bidak lawan) atau memindahkan Raja ke petak yang tidak sedang diserang. Rokade tidak diijinkan apabila Raja sedang diskak (Magethi, 2009:32).

2.2.6. Akhir Permainan Catur

Tujuan permainan adalah mencapai posisi skak mat. Hal ini bisa terjadi bila Raja terancam dan tidak bisa menyelamatkan diri ke petak lain. Tidak selalu permainan berakhir dengan kekalahan, karena bisa terjadi pula peristiwa seri atau remis di mana kedua belah pihak tidak mampu lagi meneruskan pertandingan karena tidak bisa mencapai skak mat. Peristiwa remis ini bisa terjadi berdasarkan kesepakatan maupun tidak. Salah satu contoh remis yang tidak berdasarkan kesepakatan, tetapi terjadi adalah pada keadaan remis abadi. Keadaan remis yang

lain adalah keadaan pat, dimana yang giliran melangkah tidak bisa melangkahkan buah apapun termasuk Raja, tetapi tidak dalam keadaan terancam skak. Dalam pertandingan catur pihak yang menang biasanya mendapatkan nilai 1, yang kalah 0, sedang draw 0.5 (Magethi, 2009:75).

2.3. *Microcontroller ATmega32*

Microcontroller AVR (Alf and Vegard's Risc processor) standar memiliki arsitektur 8 bit, dimana semua instruksi dikemas dalam kode 16-bit, dan sebagian besar instruksi dieksekusi dalam 1 (satu) siklus clock. AVR berteknologi RISC (Reduced Instruction Set Computing), sedangkan seri MCS51 berteknologi CISC (Complex Instruction Set Computing). AVR dapat dikelompokkan menjadi empat kelas, yaitu keluarga ATtiny, keluarga AT86RFxx, keluarga ATmega, dan keluarga AT86RFxx. Pada dasarnya, yang membedakan masing-masing kelas adalah memori, peripheral, dan fungsinya. Untuk *microcontroller AVR* yang berukuran lebih kecil, Anda dapat mencoba ATmega8 atau ATtiny2313 dengan ukuran Flash Memory 2KB dengan dua input analog (Adrianto, 2008:17).

Selain ATmega32, sangat direkomendasikan untuk mencoba ATmega16 dan ATmega28. Selain itu, kuasai juga jenis *microcontroller* lain produksi Maxim (Maxim-ic.com) seperti DS80C400 dan MAXQ2000 (Adrianto, 2008:19). Di dalam *microcontroller ATmega32* sudah terdiri dari:

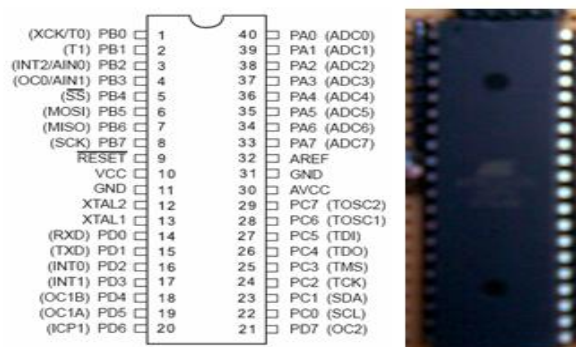
- Saluran I/O ada 32 buah, yaitu Port A, Port B, Port C, dan Port D.
- ADC (*Analog to Digital Converter*) 10 bit sebanyak 8 channel.
- Tiga buah *Timer/Counter* dengan kemampuan perbandingan.
- CPU yang terdiri dari 32 buah *register*.

- 131 instruksi andal yang umumnya hanya membutuhkan 1 siklus *clock*.
- *Watchdog timer* dengan osilator internal.
- Dua buah *timer/counter* 8 bit, satu *buah timer/counter* 16 bit.
- Tegangan operasi 2.7 V – 5.5 V pada ATmega32L.
- Internal SRAM sebesar 1 KB.
- Memori *Flash* sebesar 32 KB dengan kemampuan *Read While Write*.
- Unit interupsi *internal* dan *eksternal*.
- Port antarmuka SPI.
- EEPROM sebesar 512 byte yang dapat diprogram saat operasi.
- Antarmuka komparator analog.
- Empat channel PWM.
- 32 x 8 *general purpose register*.
- Hampir mencapai 16 MIPS pada Kristal 16 MHz.
- Port USART *programmable* untuk komunikasi serial.

2.3.1. Konfigurasi PIN

Gambar 2.2 merupakan susunan kaki standar 40 pin DIP *microcontroller*

AVR ATmega32:



Gambar 2.4. Pin-pin ATmega32

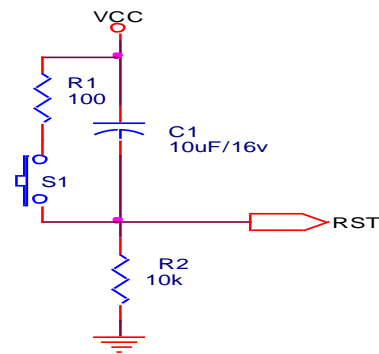
(Sumber: <http://student.eepis-its.edu/~ulopens/Tutorial/Mikrokontroler>)

Berikut penjelasan umum susunan kaki ATmega32 :

- VCC merupakan pin masukan positif catu daya. Setiap peralatan elektronika digital tentunya butuh sumber catu daya yang umumnya sebesar 5V, itulah sebabnya di PCB kit *microcontroller* selalu ada *IC regulator 7805*.
- GND sebagai pin *Ground*.
- Port.A (PA0-PA7) merupakan pin I/O dua arah dan dapat diprogram sebagai pin masukan ADC.
- Port B (PB0-PB7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu *Timer/Counter*, Komparator analog, dan SPI.
- Port C (PC0-PC7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu TWI, komparator analog, dan *Timer Osilator*.
- Port D (PD0-PD7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu komparator *analog*, interupsi eksternal dan komunikasi serial.
- *Reset* merupakan pin yang digunakan untuk *me-reset microcontroller*
- XTAL 1 dan XTAL 2 sebagai pin masukan *clock* eksternal. Suatu *microcontroller* membutuhkan sumber detak (*clock*) agar dapat mengeksekusi instruksi yang ada di memori. Semakin tinggi nilai kristalnya, maka semakin cepat *microcontroller* tersebut.
- AVCC sebagai pin masukan tegangan untuk ADC.
- AREF sebagai pin masukan tegangan referensi.

2.3.2. *Reset*

Skema rangkaian *reset* dapat dilihat pada Gambar 2.5 sebagai berikut:



Gambar 2.5. Skema rangkaian *reset*
(Sumber: Nalwan, 2003:56)

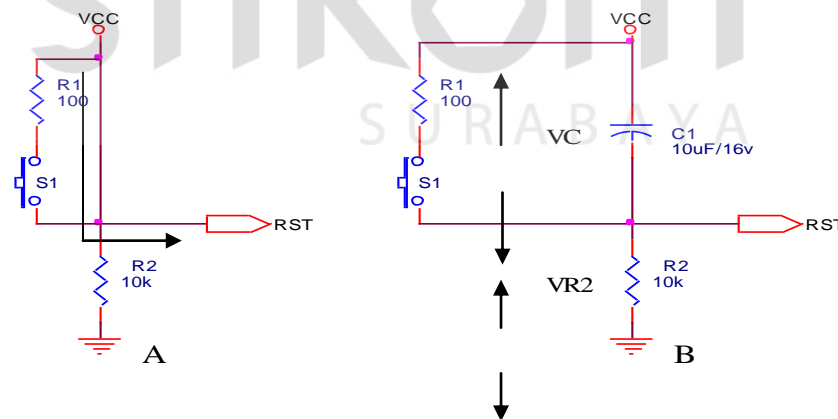
Reset dapat dilakukan secara manual maupun otomatis saat power diaktifkan (*Power On Reset*). Saat terjadi *reset* isi dari *register* akan berubah sesuai yang ada pada Tabel 2.1 (Nalwan, 2003:58).

Tabel 2.1. Isi *Register* setelah *reset*

<i>Register</i>	<i>Isi Register</i>
Program Counter	0000H
Akumulator	00H
Register B	00H
PSW	00H
Stack Pointer (A)	07H
DPTR	0000H
Port 0 – 3	FFH
Interrupt Priority (IP)	XXX0000B
Interrupt Enable (IE)	0XX0000B
Register Timer	00H
SCON	00H
SBUF	00H
PCON (HMOS)	0XXXXXXXB
PCON (CMOS)	0XXX000B

Reset terjadi dengan adanya logika 1 selama minimal 2 *cycle* pada kaki RST. Setelah kondisi pin RST kembali *low*, *microcontroller* akan mulai menjalankan program dari alamat 0000H. Kondisi pada *internal* RAM tidak terjadi perubahan selama *reset*.

Gambar 2.6 merupakan gambar rangkaian *reset* yang bekerja secara manual atau otomatis saat sumber daya diaktifkan. Saat sumber daya diaktifkan, maka kapasitor C1 sesuai dengan sifat kapasitor akan terhubung singkat pada saat itu sehingga rangkaian ekivalennya tampak pada Gambar 2.6A. Arus mengalir dari VCC langsung ke kaki RST sehingga kaki tersebut berlogika 1. Kemudian kapasitor terisi hingga tegangan pada kapasitor (VC) yaitu tegangan antara VCC dan titik antara kapasitor C1 dan resistor R2 mencapai VCC, otomatis tegangan pada R2 atau tegangan RST akan turun menjadi 0 sehingga kaki RST akan berlogika 0 (Gambar 2.6B) dan proses *reset* selesai. Aliran arus dan perubahan tegangan pada *reset* otomatis dapat dilihat pada Gambar 2.6.

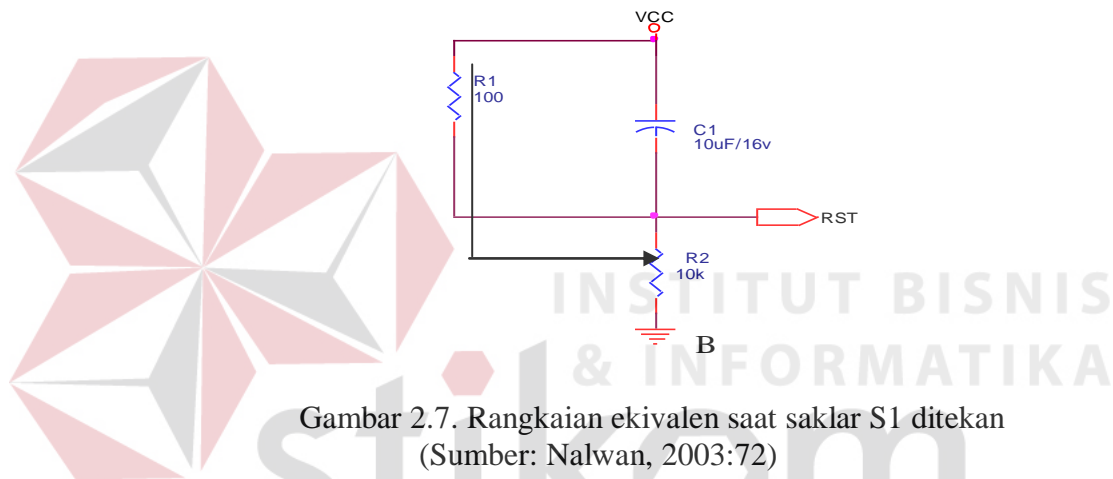


Gambar 2.6. Aliran arus dan perubahan tegangan pada *reset* otomatis
(Sumber: Nalwan, 2003:62)

Jika saklar S1 ditekan, *reset* kembali bekerja secara manual, aliran arus akan mengalir dari VCC melalui R1 menuju kaki RST (Gambar 2.6B). Tegangan pada kaki RST atau VR2 akan berubah menjadi (Nalwan, 2003:67) :

$$VR2 = \frac{R2 \times VCC}{R1 + R2} \quad (2.1)$$

Yaitu 4,95 volt dengan nilai VCC = 5 volt. Sedangkan untuk rangkaian ekivalen saat saklar S1 ditekan dapat dilihat pada Gambar 2.7 berikut:



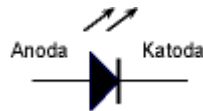
Gambar 2.7. Rangkaian ekivalen saat saklar S1 ditekan
(Sumber: Nalwan, 2003:72)

Tegangan 4,94 volt pada kaki RST menyebabkan kaki ini berlogika 1 pada saat saklar tersebut ditekan. Saat saklar dilepas, aliran arus dari VCC melalui R1 akan terhenti dan tegangan pada kaki RST akan menurun menuju nol sehingga logika pada kaki ini menjadi 0 dan proses *reset* selesai.

2.4. *Light Emitting Diode (LED)*

LED adalah singkatan dari *Light Emitting Diode*, merupakan komponen yang dapat mengeluarkan emisi cahaya. LED merupakan produk temuan lain setelah dioda. Strukturnya juga sama dengan dioda, tetapi belakangan ditemukan bahwa elektron yang menerjang sambungan P-N juga melepaskan energi berupa

energi panas dan energi cahaya. LED dibuat agar lebih efisien jika mengeluarkan cahaya. Untuk mendapatkna emisi cahaya pada semikonduktor, doping yang pakai adalah galium, arsenic dan phosphorus. Jenis doping yang berbeda menghasilkan warna cahaya yang berbeda pula (Mismail, 2007:112). Berikut adalah gambar dari simbol dioda LED :



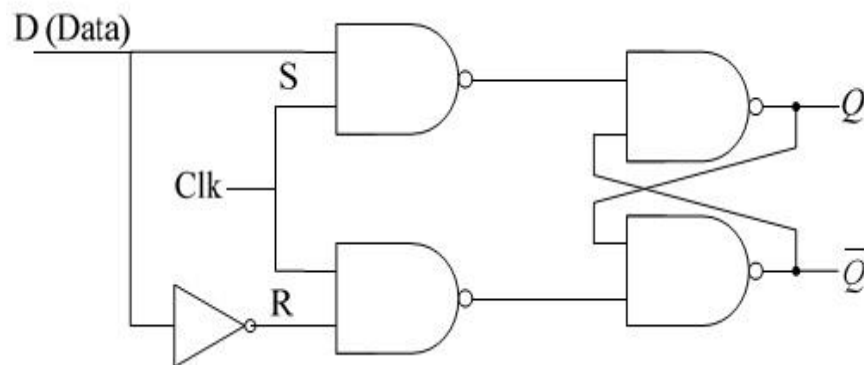
Gambar 2.8. Simbol LED

(Sumber: http://en.wikipedia.org/wiki/Light-emitting_diode)

Dalam memilih LED perlu diperhatikan tegangan kerja, arus maksimum dan disipasi daya-nya. Pembungkus (*chasing*) LED dan bentuknya juga bermacam-macam, ada yang persegi empat, bulat dan lonjong (Mismail, 2007:114).

2.5. Latch 74HC573

Latch merupakan konsep dasar elemen penyimpan dalam sistem digital. Bentuk *latch* yang secara praktis penggunaannya luas adalah *D-Latch*. Rangkaian *D-latch* ini memiliki masukan tunggal yaitu D (Data), dan akan menyimpan masukan D dengan pengendali sinyal Clk (*clock*). Rangkaian ini dikenal sebagai *Gated D latch* dengan bentuk implementasi sebagai berikut :

Gambar 2.9. *Gated D-Latch*

(Sumber: http://en.wikipedia.org/wiki/flip-flop_electronics)

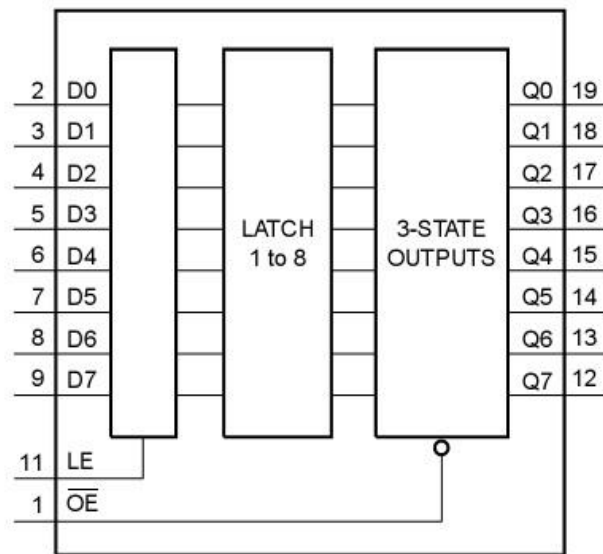
Jika $D = 1$, maka $S = 1$ dan $R = 0$, sehingga akan mengakibatkan state Q bernilai 1 ($Q = 1$). Sedangkan jika $D = 0$, maka $S = 0$ dan $R = 1$ yang berakibat pada state Q bernilai 0 ($Q = 0$). Perubahan state pada Q akan terjadi jika Clk bernilai 1, sedangkan saat Clk bernilai 0, Q akan mempertahankan state terakhirnya. Tabel kebenaran yang menggambarkan cara kerja dari *gated D latch* ini adalah sebagai berikut:

Clk	D	Q(t+1)
0	x	Q(t)
1	0	0
1	1	1

Gambar 2.10. Tabel kebenaran D-Latch

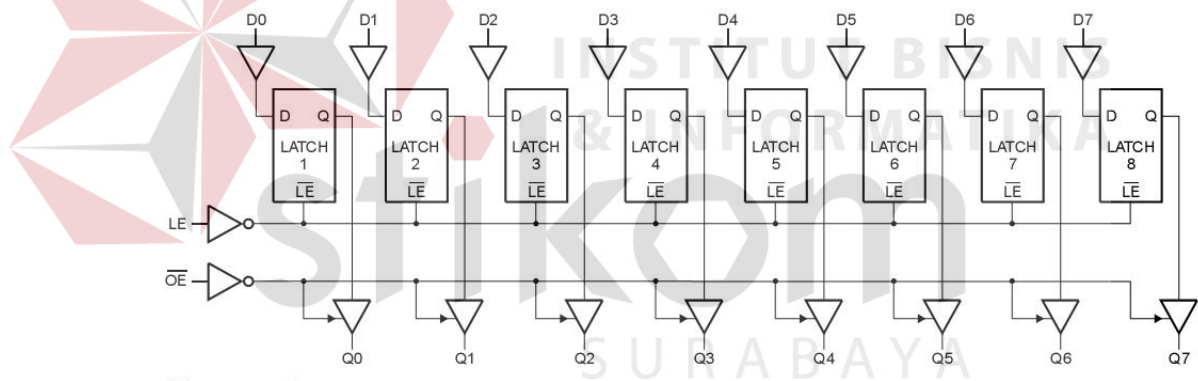
(Sumber: http://en.wikipedia.org/wiki/flip-flop_electronics)

74HC573 adalah gerbang semikonduktor berkecepatan tinggi yang memiliki tipe octal *D-latch*. IC 74HC573 ini terdiri dari 3 blok yaitu input, *D-Latch* dan 3-State output. Berikut adalah *function diagram* dari 74HC573 :



Gambar 2.11. *Diagram function 74HC57*
(Sumber: http://en.wikipedia.org/wiki/Latch_electronics)

Berikut adalah *logic diagram* dari latch 74HC573 :



Gambar 2.12. *Logic diagram 74HC573*
(Sumber: http://en.wikipedia.org/wiki/Latch_electronics)

Beberapa fitur-fitur pada *latch* 74HC573 adalah sebagai berikut :

1. Input dan output berada pada posisi berlawanan sehingga memudahkan dalam desain untuk penggunaannya bersama *microcontroller*.
2. Dapat dipergunakan sebagai input juga output pada mikroprosesor dan mikrokomputer.

3. Memiliki 3-State output non-*inverting* untuk aplikasi berorientasi bus.
4. *Common 3-state output enable input*.
5. *Functionally identical to 74HC563; 74HCT563 and 74HC373; 74HCT373.*
6. Dilengkapi JEDEC dengan standar nomor 7A.
7. *ESD protection*:
 - a. HBMEIA/JESD22-A114-C *exceeds* 2000V.
 - b. MMEIA/JESD22-A115-A *exceeds* 200V.
8. Spesifikasi dari -40 °C hingga +85 °C dan dari -40 °C hingga +125 °C.

Berikut adalah gambar dari IC *Latch* 74HC573 :



Gambar 2.13. IC *Latch* 74HC573

(Sumber: http://en.wikipedia.org/wiki/Latch_electronics)

2.6. *Buffer* 74HC244

Buffer adalah proses penampungan data sementara pada *memory* untuk keperluan penggunaan selanjutnya, proses bisa berupa digandakan, dipindahkan, diunduh, dimonitor dan lain sebagainya. Tugas utama dari *buffer* adalah untuk keperluan transfer data dari satu tempat ke tempat lain. *Buffer* adalah sebuah daerah memori yang menyimpan data ketika data tersebut ditransfer antara dua

perangkat atau antara sebuah perangkat dan sebuah aplikasi. *Buffering* digunakan karena tiga alasan, antara lain:

1. Untuk mengatasi perbedaan kecepatan antara produsen dan konsumen dari sebuah aliran data .
2. Untuk menyesuaikan antara perangkat-perangkat yang mempunyai perbedaan ukuran transfer data . Perbedaan ukuran transfer data ini sangat umum terjadi pada jaringan komputer dimana *buffer* digunakan secara luas untuk fragmentasi dan pengaturan kembali pesan-pesan. Pada bagian pengiriman, pesan yang ukurannya besar akan dipecah-pecah menjadi paket-paket kecil (fragmentasi). Paket-paket ini dikirim melalui jaringan, di ruang penerimaan, paket-paket kecil tadi diletakkan dalam *buffer* untuk disatukan kembali.
3. Untuk mendukung *copy semantic* pada aplikasi M/K. Sebuah contoh akan menjelaskan arti dari *copy semantic*. Misalkan sebuah aplikasi mempunyai *buffer* data yang ingin dituliskan ke disk. Aplikasi tersebut akan memanggil *system call write*, lalu menyediakan sebuah pointer ke *buffer* dan sebuah bilangan bulat (integer) yang menspesifikasikan jumlah byte yang ditulis. Setelah *system call* tersebut selesai, dengan *copy semantic*, versi data yang ditulis ke disk sama dengan versi data pada saat aplikasi memanggil *system call write* , tidak tergantung dengan perubahan apapun yang ada pada *buffer*. Cara sederhana sistem operasi dapat menjamin *copy semantic* adalah untuk *system call write* dengan menyalin data aplikasi ke *buffer kernel* sebelum mengembalikan kontrol ke aplikasi. Penulisan ke disk dilakukan dari *buffer kernel* sehingga perubahan yang terjadi pada *buffer* aplikasi tidak mempunyai efek apapun. Menyalin data antara *buffer kernel* dan *buffer* aplikasi adalah hal

yang umum dalam sistem operasi, kecuali *overhead* yang ada pada *clean semantic*. Efek yang sama dapat diperoleh dengan hasil yang lebih efisien dengan penggunaan yang cermat pada pemetaan memori virtual dan perlindungan halaman *copy-on-write*.

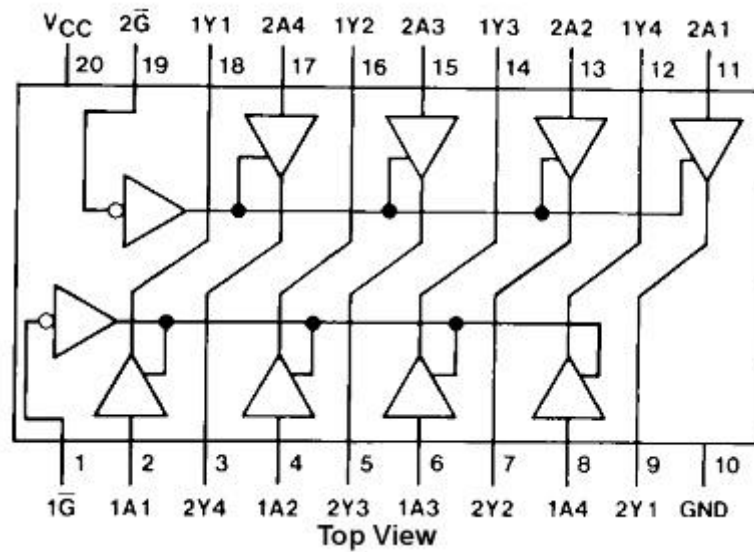
74HC244 adalah *buffer non-inverting* dan memiliki dua *active-low enable* (1G dan 2G). Setiap jalur *enable* berdiri sendiri mengendalikan empat *buffer*. CMOS 74HC244 ini tidak memiliki input *Schmit trigger*. Berikut adalah beberapa fitur yang dimiliki IC ini :

1. Memiliki delay propagasi : 14 ns.
2. Memiliki 3-State output untuk koneksi bus.
3. Rentang nilai power supply antara 2-6 V.
4. dilengkapi dengan standar JEDEC no. 7A.
5. Spesifikasi dari -40 °C hingga +85 °C dan dari -40 °C hingga +125 °C.
6. ESD *protection*:

HBM EIA/JESD22-A114-C *exceeds* 2000 V.

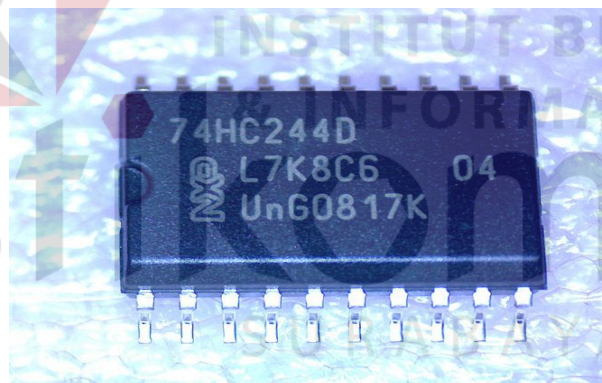
MM EIA/JESD22-A115-A *exceeds* 200 V.

Gerbang semikonduktor 74HC244 ini memiliki *output-drive* yang tinggi sehingga memungkinkan operasi kecepatan tinggi bahkan pada saat digunakan untuk bus berkapasitas besar. Semua inputan juga terlindungi dari kerusakan akibat *static-discharge* dari dioda ke Vcc dan *Ground*. Berikut adalah gambar dari diagram fungsi *buffer* 74HC244 :



Gambar 2.14. Diagram fungsi 74HC244
(Sumber: <http://en.wikipedia.org/wiki/Buffer>)

Berikut adalah gambar chip dari *buffer* 74HC244 :



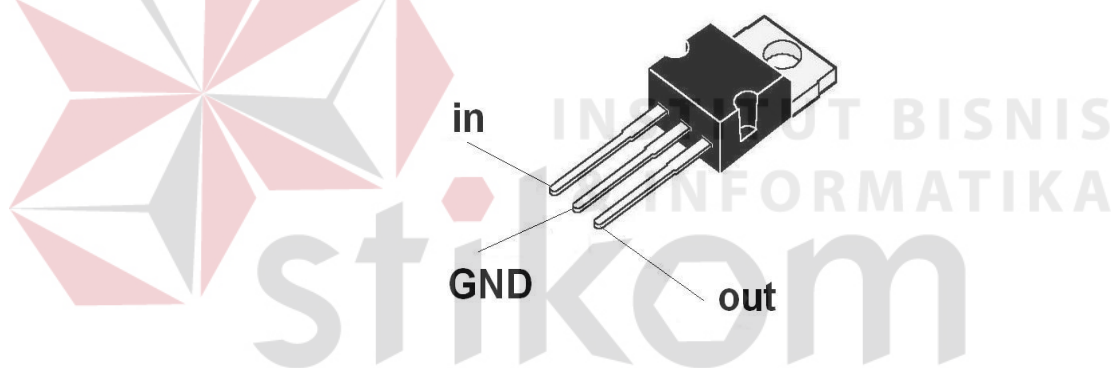
Gambar 2.15. IC *buffer* 74HC244
(Sumber: <http://en.wikipedia.org/wiki/Buffer>)

2.7. Regulator LM7805

Regulator adalah rangkaian regulasi atau pengatur tegangan keluaran dari sebuah catu daya agar efek dari naik atau turunnya tegangan jala tidak mempengaruhi tegangan catu daya sehingga menjadi stabil.

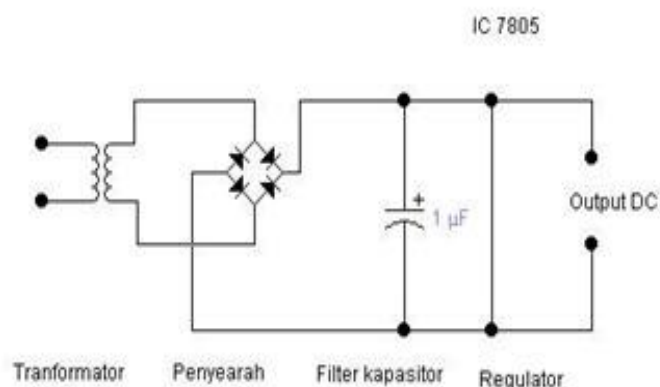
Prinsip Kerja IC (*integrated circuit*) mempunyai kelebihan karena lebih kompak (praktis) dan umumnya menghasilkan penyetabilan tegangan yang lebih baik. Fungsi-fungsi seperti pengontrol, sampling, komparator, referensi, dan proteksi yang tadinya dikerjakan oleh komponen diskrit, sekarang semuanya dirangkai dan dikemas dalam IC.

LM7805 adalah regulator untuk mendapat tegangan 5 volt. Komponen ini dilengkapi dengan pembatas arus (*current limiter*) dan juga pembatas suhu (*thermal shutdown*). Komponen ini memiliki tiga pin dan dengan menambah beberapa komponen saja sudah dapat menjadi rangkaian catu daya yang ter-regulasi dengan baik. Berikut adalah gambar dari regulator LM7805 :



Gambar 2.16. Regulator LM7805
(Sumber: <http://id.wikipedia.org/wiki/Transformator>)

Adaptor atau catu daya adalah sumber tegangan DC yang digunakan untuk memberikan tegangan akan daya kepada berbagai rangkaian elektronika yang membutuhkan tegangan DC agar dapat beroperasi. Berikut adalah skematik rangkaian lengkap catu daya menggunakan regulator tiga terminal IC 7805 untuk tegangan output 5 volt konstan.



Gambar 2.17. Skematik prinsip kerja IC regulator LM7805

Arus maksimum regulator IC yang dikirim ke beban tergantung pada tiga faktor, yaitu temperatur, arus beban, dan perbedaan antara input dan output atau yang disebut dengan diferensial input dan output (Blocher, 2003:134).

2.8. BASCOM-AVR Compiler

BASCOM-AVR adalah program Basic compiler berbasis Windows untuk *microcontroller* keluarga AVR. *Microcontroller* AVR dapat dikelompokkan menjadi 6 kelas, yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga ATmega, keluarga AT90CAN, keluarga AT90PWM dan AT89RFxx. BASCOM-AVR merupakan pemrograman dengan bahasa tingkat tinggi Basic yang dikembangkan oleh MCS *electronic*. BASCOM-AVR dapat dijalankan pada sistem operasi Windows 95, 98, NT, 2000, XP, Vista dan 7. Berikut adalah beberapa keunggulan BASCOM-AVR (Iswanto, 2008:87) :

- a. Basic yang terstruktur dengan label.
- b. Pemrograman terstruktur dengan IF-THEN-ELSE-END IF, DO-LOOP, WHILE-WEND, SELECT- CASE.
- c. Kode mesin yang cepat bukan kode yang diinterpretasikan.

- d. Variabel dan label berkapasitas besar hingga 32 karakter.
- e. Variabel yang tersedia antara lain : Bit, Byte, Integer, Word, Long, Single, Double dan String.
- f. Memberikan fasilitas untuk variabel Double. Hanya BASCOM-AVR yang dapat memberikan fasilitas ini dan tidak ditemukan di compiler AVR yang lainnya. BASCOM juga mampu menampung variabel Double dalam jumlah besar dengan 8 byte *Floating Point*.
- g. *Large set of Trig Floating point functions.*
- h. *Date & Time calculation functions.*
- i. Program compiler yang dapat berfungsi di semua *microprocessors* AVR yang memiliki *internal memory*.
- j. Sangat kompatibel dengan Microsoft VB / QB.
- k. Memiliki perintah khusus untuk LCD-display, chip I2C dan chip 1WIRE, PC Keyboard, matriks Keyboard, input RC5, perangkat lunak UART, SPI, grafis LCD, IR kirim RC5, RC6 atau kode Sony.
- l. TCP/IP dengan W3100A chip.
- m. *Local variables, user functions, library support.*
- n. *Integrated terminal emulator with download option.*
- o. *Integrated simulator for testing.*
- p. *Integrated ISP programmer (application note AVR910.ASM).*
- q. *Integrated STK200 programmer and STK300 programmer.*
- r. *Editor with statement highlighting.*
- s. *PDF datasheet viewer.*
- t. *Context sensitive help.*

- u. Sangat sesuai untuk pemakaian *board* berikut :
 - a. AVR robot controller (ARC 1.1) dari L. Barello.
 - b. Active Mega8535 Micro Board dari Active Robots.
 - c. MAVRIC dan MAVRIC-II dari BDMICRO.
- v. *Special Tcp/Ip library, AT mouse simulator, AT keyboard simulator and others are available as add on.*

2.9. CodeVision AVR

CodeVisionAVR merupakan sebuah *cross-compiler C, Integrated Development Environment (IDE)*, dan *Automatic Program Generator* yang di desain untuk *microcontroller* buatan Atmel seri AVR. CodeVisionAVR dapat dijalankan pada sistem operasi Windows 95, 98, Me, NT4, 2000, XP, Vista dan 7. *Cross-compiler C* mampu menerjemahkan hampir semua perintah dari bahasa ANSI C, sejauh yang diijinkan oleh arsitektur dari AVR, dengan tambahan beberapa fitur untuk mengambil kelebihan khusus dari arsitektur AVR dan kebutuhan pada sistem *embedded* (Adrianto, 2008:32).

File object COFF hasil kompilasi dapat digunakan untuk keperluan *debugging* pada tingkatan C, dengan pengamatan *variabel*, menggunakan *debugger* Atmel AVR Studio.

IDE mempunyai fasilitas internal berupa *software AVR Chip In-System Programmer* yang memungkinkan Anda untuk melakukan *transfer program* kedalam chip *microcontroller* setelah sukses melakukan kompilasi secara otomatis. *Software In-System Programmer* didesain untuk bekerja dengan Atmel STK500/AVRISP/AVRProg, Kanda Systems STK200+/300, Dontronics DT006,

Vogel Elektronik VTEC-ISP, Futurlec JRAVR dan MicroTronics ATCPU/Mega2000 *programmers/development boards*.

Untuk keperluan *debugging sistem embedded*, yang menggunakan komunikasi serial, IDE mempunyai fasilitas internal berupa sebuah *terminal*. Selain *library standar C*, CodeVisionAVR juga mempunyai *library* tertentu untuk:

- a. Modul LCD *alphanumeric* Bus I2C dari Philips.
- b. Sensor Suhu LM75 dari National *Semiconductor*.
- c. *Real-Time Clock*: PCF8563, PCF8583 dari Philips, DS1302 dan DS1307 dari Maxim/Dallas *Semiconductor*.
- d. Protokol 1-Wire dari Maxim/Dallas *Semiconductor*.
- e. Sensor Suhu DS1820, DS18S20, dan DS18B20 dari Maxim/Dallas *Semiconductor*.
- f. *Termometer/Termostat* DS1621 dari Maxim/Dallas *Semiconductor*.
- g. EEPROM DS2430 dan DS2433 dari Maxim/Dallas *Semiconductor*.
- h. SPI.
- i. *Power Management*.
- j. *Delay*.
- k. Konversi ke Kode Gray

CodeVisionAVR juga mempunyai *Automatic Program Generator* bernama CodeWizardAVR, yang memungkinkan Anda untuk menulis, dalam hitungan menit, semua instruksi yang diperlukan untuk membuat fungsi-fungsi berikut:

1. Set-up akses memori eksternal.

2. Identifikasi sumber *reset* untuk *chip*.
3. Inisialisasi *port input/output*.
4. Inisialisasi interupsi eksternal.
5. Inisialisasi *Timer/Counter*.
6. Inisialisasi *Watchdog-Timer*.
7. Inisialisasi UART (USART) dan komunikasi serial berbasis *buffer* yang digerakkan oleh interupsi.
8. Inisialisasi Pembanding *Analog*.
9. Inisialisasi ADC.
10. Inisialisasi Antarmuka SPI.
11. Inisialisasi Antarmuka *Two-Wire*.
12. Inisialisasi Antarmuka CAN.
13. Inisialisasi *Bus I2C*, Sensor Suhu LM75, *Thermometer/Thermostat* DS1621 dan *Real-Time Clock* PCF8563, PCF8583, DS1302, dan DS1307.
14. Inisialisasi *Bus I-Wire* dan Sensor Suhu DS1820, DS18S20.
15. Inisialisasi modul LCD.

2.10 Pemrograman Basic

Basic adalah bahasa pemrograman yang terstruktur. Selain itu bahasa ini mendukung teknik pemrograman modular atau prosedural yang ditandai dengan tersedianya fasilitas untuk membuat suatu prosedur atau sub program. Struktur program Basic diawali dengan bagian pendeklarasian variabel, kemudian bagian tubuh program tempat meletakkan *statement* atau instruksi-intruksi untuk sebuah

program dan diakhiri oleh *statement* END. Jika dibuat contoh terlihat sebagai berikut:

```

DIM nama_variabel AS tipe_data
...
Variabel
No_baris1 Statement_1
No_baris2 Statement_2
...
No_barisn Statement_n
End

```

Statement END tidak harus berada di akhir baris program, fungsinya adalah untuk mengakhiri program dan kembali ke sistem operasi.

Variabel adalah besaran atau simbol yang digunakan untuk menyimpan suatu nilai. Basic memiliki aturan atau ketentuan untuk penulisan nama variabel yang akan dideklarasikan, yaitu (Iswanto, 2008:23) :

1. Maksimum panjangnya 40 karakter.
2. Terdiri dari huruf, angka, dan titik, tetapi karakter pertama harus huruf.
3. Tidak boleh menggunakan spasi atau blank di antara karakter-karakter.
4. Tidak boleh ada karakter khusus, kecuali %, !, # dan \$ yang harus diletakkan diakhir nama variabel yang menunjukkan jenis variabelnya.
5. Tidak boleh sama dengan Basic *reserved words*, yaitu kata-kata yang sudah menjadi milik Basic untuk tugas tertentu. *Reserved word* terdiri dari *statement*, *function*, *keyword* dan *meta command*.

Tipe data adalah jenis data yang disimpan variabel. Jika dideklarasikan tanpa menyebutkan tipe datanya Basic menganggap variabel tersebut bertipe numerik. Tipe data dalam Basic, yaitu (Iswanto, 2008:25) :

1. Integer : bilangan bulat 16-bit antara -32768 s/d. 32767.
2. Long : bilangan bulat 32-bit.

3. Single : floating point 32-bit antara 2.938736×10^{-39} s/d 1.701412×10^{38} .
4. Double : floating point 64-bit antara $2.9358745877055719 \times 10^{-39}$ s/d $1.701411834604692 \times 10^{38}$.
5. String : untaian kata yang terdiri dari huruf dan angka dengan panjang n bytes(karakter).

