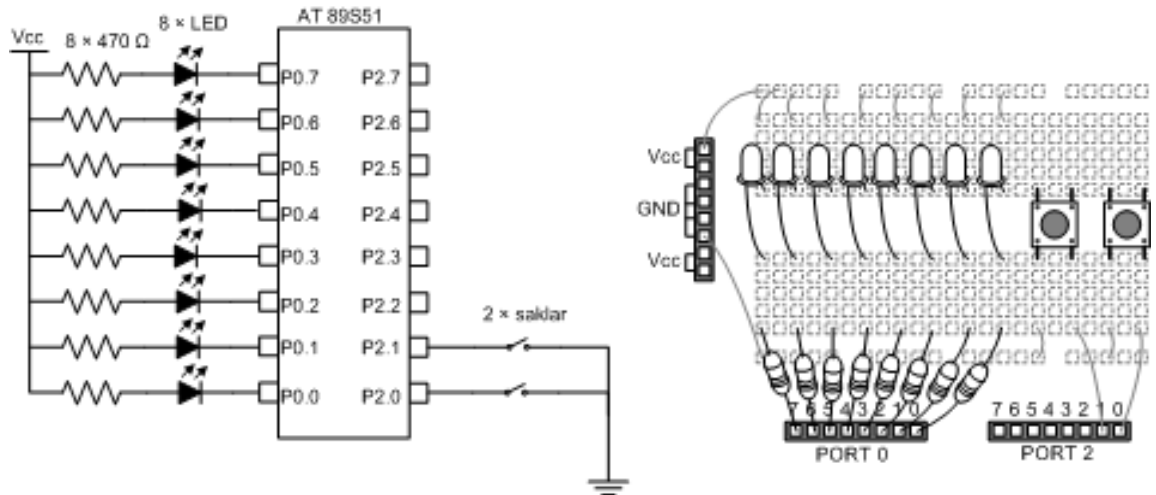


Belajar Pemrograman Mikrokontroler AT89S51 Menggunakan Bascom-8051

Modul Kuliah Kontrol Berbasis Komputer
Versi 0.2



Disusun Oleh :
Agung Nugroho Adi

Laboratorium Mekatronika
Jurusan Teknik Mesin
Universitas Islam Indonesia
2007

Daftar Isi

1. Pengantar Mikrokontroler	1
2. Dasar Pemrograman Bascom-8051	9
3. Output LED.....	15
4. Input Saklar.....	21
5. Modul Display 7 Segment	25
6. Display 7 Segment Menggunakan Rangkaian Multipleks.....	29
7. LCD	33
8. Keypad	41
9. ADC.....	45
10. Komunikasi Serial Sinkron	49
11. Komunikasi Serial Asinkron Dengan PC.....	51
12. Aktuator Motor DC	57

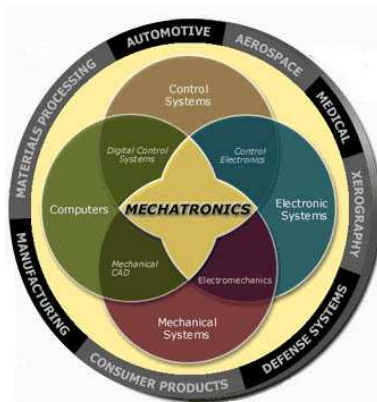
1. Pengantar Mikrokontroler

Tujuan : Setelah mempelajari bab ini mahasiswa diharapkan dapat :

- Membedakan antara mikrokontroler dengan mikroprosesor
- Menjelaskan komponen utama yang terdapat pada sistem berbasis mikroprosesor
- Menerangkan dan memberi contoh mengenai embedded system dan dedicated system
- Menerangkan urutan langkah pemrograman mikrokontroler

Batas formal antara berbagai disiplin ilmu rekayasa saat ini semakin kabur seiring dengan perkembangan teknologi IC (rangkaiannya elektronika terpadu) dan komputer. Hal ini terutama terlihat jelas pada bidang mekanik dan elektronik dimana semakin banyak produk yang merupakan integrasi dari komponen mekanik dan elektronik. Bidang mekatronika telah berkembang untuk memperluas cakupan bidang elektromekanik seperti dapat diilustrasikan pada gambar 1-1. Terdapat banyak definisi yang dapat menerangkan definisi mekatronika. Salah satunya yaitu Alciatore menyatakan (Alciatore, 2003)

Mekatronika dapat didefinisikan sebagai bidang studi yang membahas analisis, desain, sintesis, dan pemilihan sistem yang mengintegrasikan komponen mekanik dan elektronik yang dikendalikan oleh sistem kendali modern, terutama yang berbasis mikroprosesor.



Gambar 1-1 Ilustrasi sistem mekatronika sebagai perpotongan dari sistem mekanik, elektronik, kontrol, dan komputer (Craig, 2001)

Saat ini sistem berbasis mikroprosesor telah berkembang mencakup berbagai macam jenis sistem kontrol, yang secara umum disebut sebagai komputer. Terdapat bermacam tingkatan komputer, tiga jenis yang paling banyak dipergunakan adalah komputer personal (*Personal Computer*), mikrokontroler, dan *Programmable Logic Controller* (PLC). Ketiganya mempunyai penggunaan yang berbeda-beda. Komputer personal banyak dipergunakan sebagai pengolah kata, *spreadsheet*, pengolahan basis data, *drafting*, dan desain grafis. Mikrokontroler dapat dianggap komputer dengan ukuran dan kapasitas kecil banyak dipakai dalam berbagai macam produk sebagai sistem yang sering disebut sebagai *embeded system*. PLC lebih banyak penggunaannya dalam aplikasi industri karena kemudahan dalam pemrograman dan wiring, serta ketahanan terhadap kondisi yang terdapat pada lingkungan mesin industri.

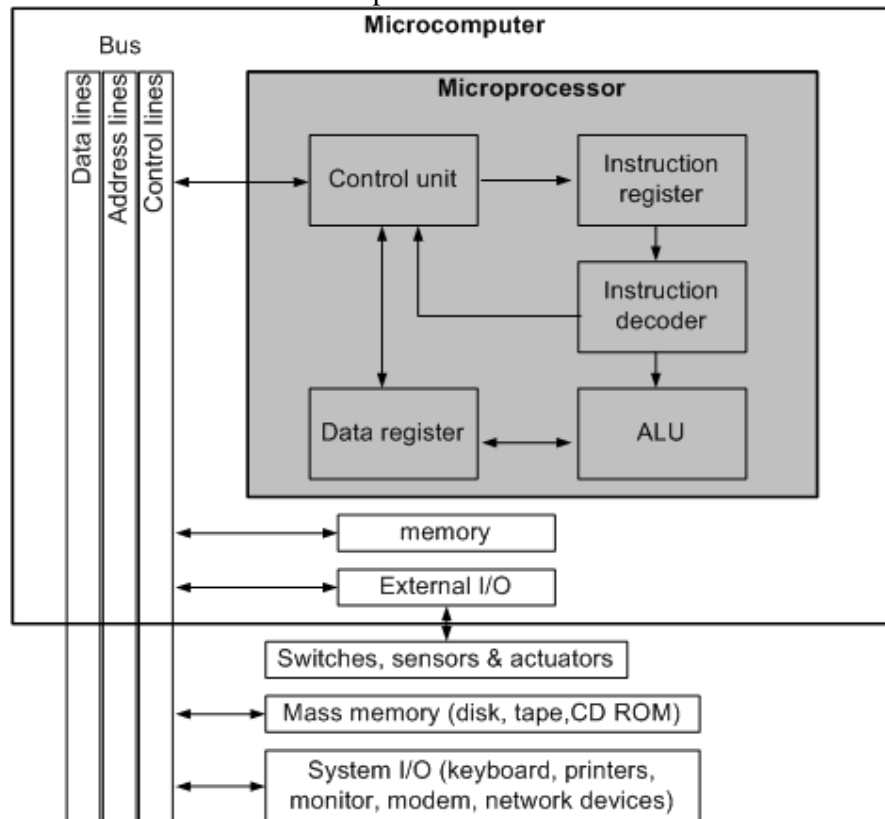
1.1 Mikroprosesor

Mikroprosesor adalah suatu chip tunggal yang memuat sirkuit digital untuk melakukan fungsi aritmetika, logika, komunikasi & kontrol dengan melakukan eksekusi instruksi (program) dalam urutan tertentu. Berbeda dengan rangkaian perangkat keras elektronik seperti rangkaian IC digital atau relay, sistem berbasis mikroprosesor mempunyai kelebihan yaitu fleksibilitas yang tinggi, karena perubahan dapat dilakukan dengan mengubah pemrograman pada perangkat lunak.

Nama lain dari mikroprosesor adalah Microprocessor Unit (MPU) atau Central Processing Unit (CPU). Contoh mikroprosesor yang paling terkenal adalah produksi Intel antara lain 6502, 8086/88, 80286, 80386, 80486, Pentium I, Pentium II, hingga generasi Dual Core. Selain intel beberapa perusahaan lain juga mengeluarkan berbagai jenis mikroprosesor mulai dari yang klasik seperti Zilog Z80, keluarga Motorola 6800 hingga generasi terbaru dari AMD.

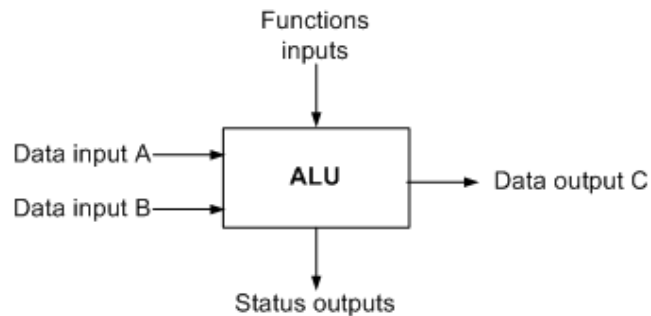
1.2 Komponen Sistem Berbasis Mikroprosesor

Supaya mikroprosesor dapat bekerja maka diperlukan komponen-komponen lainnya. Gambar 1-2 memperlihatkan komponen dari suatu mikrokomputer sebagai representasi dari sistem berbasis mikroprosesor.



Gambar 1-2 Komponen utama suatu mikrokomputer

Mikroprosesor merupakan pusat pengolahan data dan pengendali utama dari suatu mikrokomputer. Control unit berfungsi untuk mengontrol komponen lainnya sehingga instruksi dapat dijalankan dalam urutan yang benar. Arithmetic Logic Unit (ALU) bertugas untuk melaksanakan fungsi matematika dan logika dari data yang berbentuk biner. Data biner tersebut diproses dalam bentuk suatu kumpulan bit yang disebut word, yang biasanya berjumlah 8, 16, 32, atau 64 bit.



Gambar 1-3 Arithmetic Logic Unit

Selain control unit dan ALU, pada mikroprosesor juga terdapat register, yaitu register data dan register instruksi. Register adalah sarana memori kecil yang dapat menerima, menahan, dan mentransfer data. Instruction decoder bertugas menerjemahkan instruksi yang diambilkan secara berurutan dari memori oleh control unit untuk disimpan oleh instruction register. Masing-masing instruksi berupa kumpulan bit kode yang memerintahkan ALU untuk melakukan manipulasi bit pada word yang disimpan pada register data. Hasil dari ALU kembali akan disimpan pada register data untuk kemudian dikirimkan ke memori oleh control unit.

Untuk menghubungkan antara mikroprosesor dan komponen lain diperlukan jalur komunikasi yang disebut bus. Bus dapat dianalogikan sebagai jaringan saraf pada tubuh manusia. Setiap komponen pada mikrokomputer terhubung dengan bus sebagai jalur pertukaran informasi. Terdapat tiga jenis bus yaitu jalur data (data bus), jalur alamat (address bus), dan jalur kendali (control bus). Data bus bertugas untuk mempertukarkan data dari dan ke berbagai komponen dalam mikrokomputer seperti mikroprosesor, memori dan peranti I/O. Address bus digunakan sebagai penunjuk peranti atau lokasi data pada memori tempat data yang akan dibaca/ditulis. Control bus membawa perintah kontrol, misal baca atau tulis, sinyal detak dan interupsi.

Memori digunakan sebagai media penyimpanan data maupun program (kumpulan instruksi). Pada dasarnya terdapat beberapa jenis memori yakni RAM (Random Access Memory) dan ROM (Read Only Memory). RAM dapat dibaca atau ditulis oleh mikroprosesor, namun data akan hilang jika catu daya dilepas sehingga disebut sebagai volatile memory. Pada saat beroperasi biasanya data dari berbagai media penyimpan akan disimpan sementara di RAM sebelum diproses pada mikroprosesor. Hal ini disebabkan kecepatan RAM yang jauh lebih tinggi dibanding dengan media penyimpanan lainnya. ROM digunakan untuk penyimpanan permanen, yang dapat dibaca oleh mikroprosesor namun mikroprosesor tidak dapat menuliskan data ke ROM. ROM disebut juga sebagai nonvolatile memory karena akan data tidak akan hilang jika dilepaskan dari catu daya.

Terdapat berbagai jenis ROM di antaranya EPROM (Erasable Programmable ROM) dan EEPROM (Electrically Erasable Programmable ROM). Data pada EPROM dapat dihapus dengan menggunakan bantuan sinar ultraviolet untuk kemudian ditulis ulang dengan data baru. Pada EEPROM penghapusan data dilakukan secara elektronik sehingga lebih mudah. Untuk penyimpanan data secara permanen dapat juga digunakan disket, CD-ROM, dan flash memory.

1.3 Mikrokontroler

Mikrokontroler pada dasarnya adalah komputer dalam satu chip, yang di dalamnya terdapat mikroprosesor, memori, jalur Input/Output (I/O) dan perangkat pelengkap lainnya. Kecepatan pengolahan data pada mikrokontroler lebih rendah jika dibandingkan dengan PC. Pada PC kecepatan mikroprosesor yang digunakan saat ini telah mencapai orde GHz,

4 Belajar Pemrograman Mikrokontroler AT89S51 Menggunakan Bascom-8051

sedangkan kecepatan operasi mikrokontroler pada umumnya berkisar antara 1 – 16 MHz. Begitu juga kapasitas RAM dan ROM pada PC yang bisa mencapai orde Gbyte, dibandingkan dengan mikrokontroler yang hanya berkisar pada orde byte/Kbyte.

Meskipun kecepatan pengolahan data dan kapasitas memori pada mikrokontroler jauh lebih kecil jika dibandingkan dengan komputer personal, namun kemampuan mikrokontroler sudah cukup untuk dapat digunakan pada banyak aplikasi terutama karena ukurannya yang kompak. Mikrokontroler sering digunakan pada sistem yang tidak terlalu kompleks dan tidak memerlukan kemampuan komputasi yang tinggi.

Sistem yang menggunakan mikrokontroler sering disebut sebagai embedded system atau dedicated system. Embedded system adalah sistem pengendali yang tertanam pada suatu produk, sedangkan dedicated system adalah sistem pengendali yang dimaksudkan hanya untuk suatu fungsi tertentu. Sebagai contoh printer adalah suatu embedded system karena di dalamnya terdapat mikrokontroler sebagai pengendali dan juga dedicated system karena fungsi pengendali tersebut berfungsi hanya untuk menerima data dan mencetaknya. Hal ini berbeda dengan suatu PC yang dapat digunakan untuk berbagai macam keperluan, sehingga mikroprosesor pada PC sering disebut sebagai general purpose microprocessor (mikroprosesor serba guna). Pada PC berbagai macam software yang disimpan pada media penyimpanan dapat dijalankan, tidak seperti mikrokontroler hanya terdapat satu software aplikasi.

Penggunaan mikrokontroler antara lain terdapat pada bidang-bidang berikut ini.

- Otomotif : Engine Control Unit, Air Bag, fuel control, Antilock Braking System, sistem pengaman alarm, transmisi otomatis, hiburan, pengkondisi udara, speedometer dan odometer, navigasi, suspensi aktif
- perlengkapan rumah tangga dan perkantoran : sistem pengaman alarm, remote control, mesin cuci, microwave, pengkondisi udara, timbangan digital, mesin foto kopi, printer, mouse.
- pengendali peralatan di industri.
- robotika.

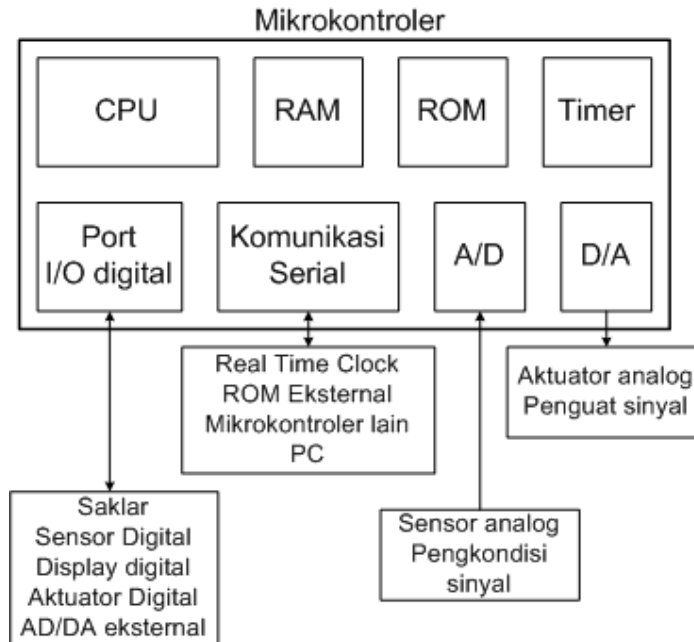
Saat ini mikrokontroler 8 bit masih menjadi jenis mikrokontroler yang paling populer dan paling banyak digunakan. Maksud dari mikrokontroler 8 bit adalah data yang dapat diproses dalam satu waktu adalah 8 bit, jika data yang diproses lebih besar dari 8 bit maka akan dibagi menjadi beberapa bagian data yang masing-masing terdiri dari 8 bit. Contoh mikrokontroler 8 bit antara lain keluarga Motorola 68HC05/11, Intel 8051, Microchip PIC 16, dan yang akhir-akhir ini mulai populer keluarga Atmel AVR. Selain yang telah disebutkan di atas terdapat juga beberapa seri mikrokontroler lain yang cukup dikenal antara lain Basic Stamp dari Parallax (banyak digunakan untuk pembelajaran mikrokontroler) dan HD64180 dari Hitachi (sebagai pengendali LCD). Masing-masing mikrokontroler mempunyai cara dan bahasa pemrograman yang berbeda, sehingga program untuk suatu jenis mikrokontroler tidak dapat dijalankan pada jenis mikrokontroler lain. Untuk memilih jenis mikrokontroler yang cocok dengan aplikasi yang dibuat terdapat tiga kriteria yaitu

- Dapat memenuhi kebutuhan secara efektif & efisien. Hal ini menyangkut kecepatan, kemasan/packaging, konsumsi daya, jumlah RAM dan ROM, jumlah I/O dan timer, harga per unit
- Bahasa pemrograman yang tersedia
- Kemudahan dalam mendapatkannya

1.3.1 Komponen Mikrokontroler

Gambar 1-4 menunjukkan komponen-komponen dari suatu mikrokontroler yang mempunyai fasilitas lengkap beserta peranti eksternal yang biasanya dihubungkan ke/dari mikrokontroler. Tidak semua mikrokontroler mempunyai semua komponen tersebut,

misalnya konverter A/D dan D/A hanya terdapat pada beberapa jenis mikrokontroler tertentu.



Gambar 1-4 Komponen dari suatu mikrokontroler

CPU pada mikrokontroler berupa mikroprosesor yang berfungsi sebagai otak dari mikrokontroler. Dalam suatu mikrokontroler biasanya terdapat tiga buah memori, yaitu RAM, ROM dan EEPROM. RAM dan ROM hampir selalu ada pada setiap mikrokontroler, sedangkan EEPROM hanya terdapat pada beberapa jenis mikrokontroler tertentu. RAM digunakan sebagai penyimpan data sementara yang berupa register-register. Register adalah tempat penyimpanan data yang berkaitan dengan banyak hal, misalnya variabel dalam program, keadaan input/output, serta pengaturan timer/counter dan komunikasi serial. Telah disebutkan sebelumnya data pada RAM akan hilang saat catu daya dicabut.

ROM digunakan sebagai tempat penyimpanan program. ROM yang banyak dipakai pada mikrokontroler saat ini adalah flash PEROM (Programmable Erasable ROM), yang mirip seperti memori pada flash disk, namun bedanya adalah flash PEROM hanya dapat dihapus dan ditulis secara sekaligus. EEPROM biasanya digunakan untuk menyimpan data yang tidak boleh hilang meski catu daya dihapus. Meski fungsinya mirip EEPROM biasanya lebih sedikit digunakan dibanding RAM karena kecepatan akses EEPROM yang lebih lambat. Contoh penggunaannya adalah penyimpanan data password, atau setting suatu sistem.

Timer/counter adalah peranti untuk mencacah sinyal dari clock ataupun sinyal dari suatu kejadian. Jika sinyal yang dicacah berasal dari clock maka peranti ini berfungsi sebagai pewaktu, sedangkan jika berasal dari clock maka peranti ini berfungsi sebagai pencacah. Pewaktu bisa digunakan untuk bermacam-macam kegunaan, misalnya untuk menghasilkan tundaan waktu dan untuk mengukur selang waktu suatu proses.

Peranti antarmuka ke input/output pada mikrokontroler disebut sebagai port. Pada satu port I/O digital terdiri beberapa pin, biasanya berjumlah 8 atau satu byte, dengan masing-masing pin dapat mentransfer satu bit data biner (logika 0 dan 1) dari/ke mikrokontroler. Selain port I/O digital, pada suatu mikrokontroler juga dapat berkomunikasi dengan peranti lain menggunakan komunikasi serial. Terdapat berbagai standar atau protokol untuk komunikasi serial seperti SPI (Serial Peripheral Interface), I2C

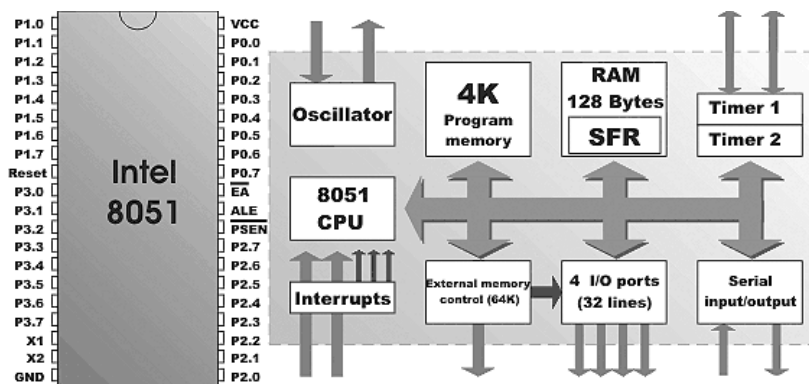
6 Belajar Pemrograman Mikrokontroler AT89S51 Menggunakan Bascom-8051

(Inter-Integrated Circuit), 1-wire, 2-wire, UART (Universal Asynchronous Receiver Transmitter) dan USART (Universal Synchronous- Asynchronous Receiver Transmitter).

1.3.2 Mikrokontroler Seri AT89S51

Mikrokontroler seri AT89S51 produksi ATMEL adalah pengembangan dari seri 8051 dari Intel. Selain ATMEL terdapat beberapa perusahaan lain yang juga mengembangkan seri 8051 ini seperti Phillips, Siemens, dan Dallas Semiconductor. Gambar 1-5 menunjukkan susunan pin dan arsitektur dari keluarga 8051.

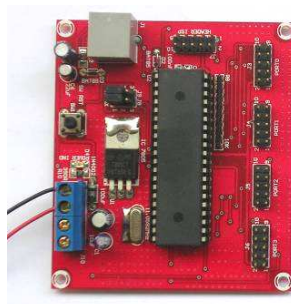
AT89S51 merupakan mikrokontroler 8 bit 40 pin dengan kapasitas RAM 128 byte dan ROM 4 kbyte serta mempunyai dua pewaktu/pencacah (timer/counter) 16 bit, satu port serial, enam sumber interupsi dan empat port input/output (I/O). Masing-masing port I/O terdiri dari 8 pin jalur data sehingga jumlah total pin I/O adalah 32 (Atmel 2001). Keluarga AT89S mempunyai beberapa seri lain selain seri AT89S51 yaitu jenis 40 pin (AT89S52/53/55) dan jenis 20 pin (AT89S2051/4051).



Gambar 1-5 Susunan pin dan arsitektur keluarga 8051

Keluarga AT89S adalah penerus dari keluarga AT89C dengan tambahan kemampuan untuk dapat diprogram secara serial. Hal ini mendukung fasilitas ISP (In System Programming) yang memungkinkan mikrokontroler untuk diprogram secara langsung tanpa memerlukan rangkaian pemrogram khusus. Hal ini berbeda dengan keluarga AT89C yang memerlukan rangkaian pemrogram tersendiri dengan dua level tegangan, yaitu 5 dan 12 Volt.

Beberapa istilah yang merujuk pada rangkaian mikrokontroler adalah rangkaian programmer/downloader dan sistem minimum. Sistem minimum adalah rangkaian dasar yang diperlukan mikrokontroler untuk dapat bekerja. Rangkaian programmer/downloader adalah rangkaian yang digunakan untuk memprogram mikrokontroler. Untuk seri AT89S51 rangkaian sistem minimumnya dapat sekaligus sebagai rangkaian programmer/downloader.

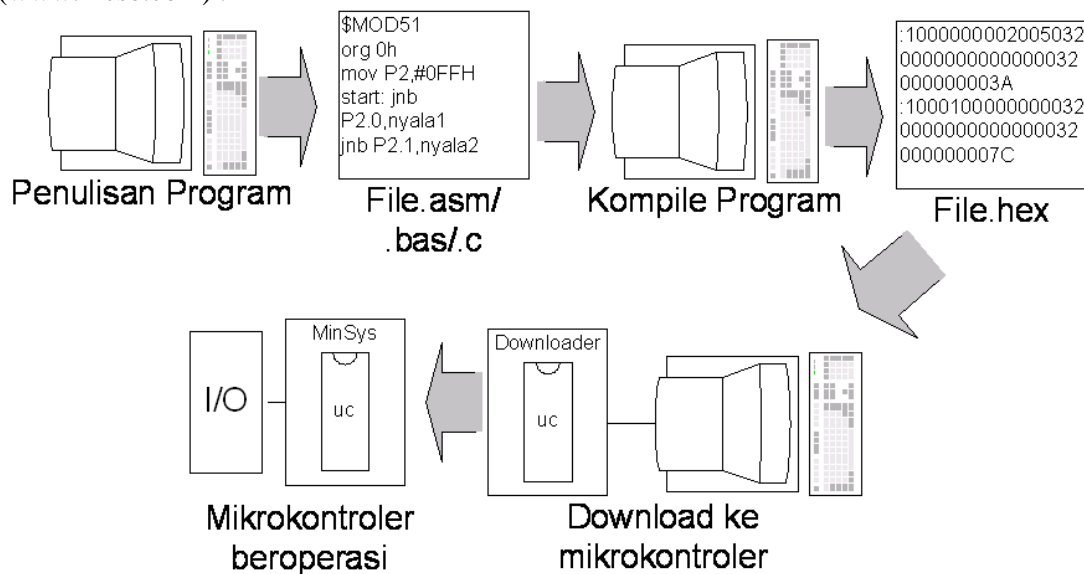


Gambar 1-6 Rangkaian sistem minimum sekaligus downloader untuk AT89S51

Gambar 1-6 menunjukkan salah satu contoh rangkaian sistem minimum untuk AT89S51. Beberapa komponen pelengkap pada sistem minimum ini adalah

- regulator untuk menjaga tegangan catu daya 5 V yang diperlukan mikrokontroler
- kristal sebagai sumber detak (clock)
- rangkaian reset
- soket untuk koneksi ke peranti I/O
- koneksi ke port paralel untuk pemrograman dari PC

Untuk memprogram suatu mikrokontroler terdapat banyak bahasa pemrograman yang dapat digunakan. Bahasa pemrograman yang biasa digunakan dalam pemrograman mikrokontroler adalah Assembly. File bahasa Assembly (ASM) dapat dituliskan menggunakan sembarang pengolah kata (misal Notepad), untuk kemudian dikompil menggunakan Assembler (misal ASM51) untuk mendapatkan file HEX. File HEX inilah yang dimasukkan ke mikrokontroler menggunakan Microcontroller ISP Software dari Atmel (ataupun software ISP lain) melalui perantara kabel paralel. Gambar 1-7 menunjukkan langkah-langkah pada pemrograman mikrokontroler. Saat ini telah dikembangkan beberapa kompilator untuk beberapa bahasa pemrograman tingkat tinggi yang dapat digunakan pada pemrograman mikrokontroler AT89S51 seperti Keil Compiler dari Keil Corp, SDCC (Small Device C Compiler) dari Sandepp Duta (<http://sdcc.sourceforge.net>) dan Bascom-8051 (Basic Compiler) dari MCS Electronics (www.mcse.com).



Gambar 1-7 Langkah-langkah pemrograman mikrokontroler.

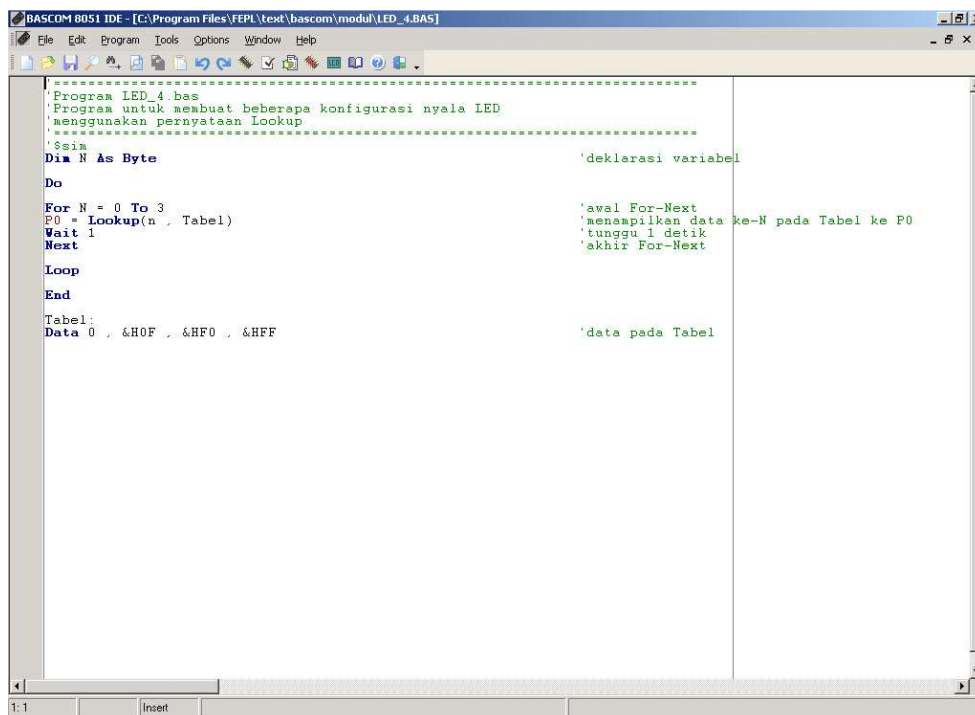
8 Belajar Pemrograman Mikrokontroler AT89S51 Menggunakan Bascom-8051

2. Dasar Pemrograman Bascom-8051

Tujuan : Setelah mempelajari bab ini mahasiswa diharapkan dapat :

- Menjelaskan dasar-dasar dan melakukan pemrograman mikrokontroler menggunakan Bascom

Bascom-8051 adalah program kompiler menggunakan Basic berbasis Windows yang dapat digunakan untuk mikrokontroler keluarga 8051, misalnya AT89S51/52/55 dan AT89S2051/4051. Versi demo Bascom-8051 yang dikembangkan oleh MCS Electronic ini dapat diunduh di www.mcselec.com secara bebas. Untuk versi demo kode yang dapat dibuat dan dijalankan mikrokontroler dibatasi besarnya maksimal 4 kByte, namun hal ini tidaklah menjadi masalah karena sesuai dengan kapasitas penyimpanan program internal pada AT89S51.



```

-----
'Program LED_4.bas
'Program untuk membuat beberapa konfigurasi nyala LED
'menggunakan pernyataan Lookup
-----
'Sisip
Dim N As Byte                                'deklarasi variabel
Do
For N = 0 To 3
PO = Lookup(n , Tabel)                       'awal For-Next
Wait 1                                       'menampilkan data ke-N pada Tabel ke P0
Next                                         'tunggu 1 detik
Loop                                         'akhir For-Next
End
Tabel:
Data 0 , &H0F , &HF0 , &HFF                'data pada Tabel

```

Gambar 2-1 Tampilan Bascom-8051

Pada umumnya bahasa yang dipergunakan untuk memprogram mikrokontroler adalah bahasa Assembly. Bahasa Assembly adalah bahasa pemrograman tingkat menengah, dimana program yang dibuat lebih mendekati bahasa mesin, sehingga pemanfaatan memori dapat dilakukan secara optimal, namun di sisi lain pemrogramannya menjadi relatif sulit.

Karena bahasa yang dipergunakan Bascom, yaitu Basic, adalah bahasa tingkat tinggi, maka pemrograman menggunakan Bascom sangatlah mudah untuk dipelajari. Sintaksnya tidak jauh berbeda dari Basic pada umumnya, misalnya do-loop, for-next, while-wend, goto, gosub dan sebagainya. Selain itu Bascom dilengkapi dengan fungsi-fungsi khusus, misalnya LCD untuk menampilkan karakter pada LCD, PRINT untuk mengirimkan karakter ke PC melalui kabel RS232, SHIFTIN dan SHIFTOUT untuk komunikasi serial sinkron dan lain sebagainya. Fungsi-fungsi khusus tersebut jika dituliskan dalam bahasa Assembly akan menjadi lebih panjang dan rumit, terutama karena kita harus mengetahui register-register yang ada pada mikrokontroler.

2.1 Karakter Pada Bascom

Karakter pada Bascom dipergunakan untuk membentuk label, keyword, variabel, dan operator, yang kesemuanya akan membentuk suatu program. Pada dasarnya karakter pada Bascom terdiri dari karakter huruf (A-Z) dan karakter angka (0-9). Beberapa karakter pada Bascom yang dipergunakan secara khusus terdapat pada tabel 2-1.

Tabel 2-1 Karakter khusus pada Bascom

Karakter	Deskripsi
ENTER	Ganti baris
	Blank (spasi)
'	Single quotation mark (apostrophe)
*	Asterisks (simbol perkalian)
+	Plus sign (simbol penjumlahan)
,	Comma
-	Minus sign (simbol pengurangan)
.	Period (decimal point)
/	Slash (simbol pembagian) ditangani seperti \
:	Colon
"	Double quotation mark
;	Semicolon
<	Kurang dari
=	Sama dengan
>	Lebih dari
\	Backslash (simbol pembagian integer)

2.2 Tipe Data

Setiap variabel pada Bascom mempunyai tipe data yang menunjukkan kapasitas dan jenis data yang dapat disimpan pada variabel tersebut. Hal ini berpengaruh pada seberapa besar memori yang diperlukan untuk menyimpan variabel tersebut. Tabel 2-2 menunjukkan tipe data pada Bascom beserta ukuran dan rentangnya.

Tabel 2-2 Tipe data pada Bascom

Tipe Data	Ukuran (byte)	Rentang
Bit	1/8	0 atau 1
Byte	1	0 – 255
Integer	2	-32768 – +32767
Word	2	0 – 65535
Long	4	-2147483648 – +2147483647
Single	4	
String	maksimum 254 byte	

2.3 Variabel

Variabel adalah simbol yang digunakan untuk mewakili suatu nilai. Variabel digunakan sebagai tempat penyimpanan data atau penampung data sementara. Variabel numerik

hanya dapat diisi nilai numerik (bit, byte, integer, word, long, dan single). Isi dari suatu variabel numerik dapat berupa :

- Suatu nilai konstan
 - A = 5
 - C = 1.1
- Nilai variabel numerik lain
 - A = B
- Nilai yang didapat dengan mengkombinasikan variabel, konstan, dan operator
 - Temp = A + 5

Pada Bascom terdapat beberapa aturan mengenai penamaan suatu variabel, yaitu :

- Nama suatu variabel maksimum terdiri atas 32 karakter dan dapat berupa huruf ataupun angka.
- Karakter pertama variabel haruslah berupa huruf.
- Nama variabel tidak boleh menggunakan kata-kata yang dipergunakan Bascom sebagai perintah, pernyataan, register dan operator (misal AND, OR, DIM, P1, TIMER0 dan lain sebagainya)

Sebelum digunakan suatu variabel haruslah dideklarasikan terlebih dahulu tipe data yang dipergunakan dengan menggunakan pernyataan DIM

```
DIM A As Byte
DIM Nama1 As Byte, Nama2 As Integer
DIM Kata As String*10
```

Selain menggunakan DIM, variabel dapat juga ditentukan tipe datanya menggunakan DEFBYTE, DEFINT, DEFBIT, dan DEFWORD

```
DEFBYTE A
DEFWORD B;C;D
```

Suatu variabel dapat mempunyai nama lain atau alias. Umumnya alias digunakan untuk mengganti variabel standar dengan nama yang lebih mudah diingat. Hal ini akan berguna pada saat membuat program yang panjang dan kompleks, jika terdapat perubahan penggunaan pin atau port, cukup diganti pada pernyataan Alias.

```
Saklar1 Alias P0.0
LED1 Alias P2.0
```

2.4 Konstanta

Konstanta adalah variabel yang mempunyai nilai konstan selama program dijalankan. Untuk mendeklarasikan suatu konstanta dapat digunakan dua cara, yaitu menggunakan Dim atau Const

```
Dim A As Const 5
Dim B1 As Const &B1001
Const Cbyte = &HF
Const Cint = -1000
Const Csingle = 1.1
Const Cstring = "tes"
```

2.5 Larik

Larik atau array adalah kumpulan variabel dengan nama dan tipe data yang sama. Untuk membedakan satu variabel dengan variabel lainnya digunakan indeks. Indeks haruslah berupa angka dengan tipe data byte, integer atau word, dengan nilai minimal 1 (bukan 0). Pendeklarasian larik mirip seperti variabel biasa, hanya ditambahkan jumlah komponen lariknya.

```
Dim a(10) as byte
```

2.6 Sistem Bilangan

Pada pemrograman mikrokontroler terdapat 3 sistem bilangan yang sering digunakan, yaitu desimal (basis 10), biner (basis 2), dan heksadesimal (basis 16). Cara penulisan bilangan pada Bascom disesuaikan dengan sistem bilangan yang digunakan, yaitu :

- Untuk bilangan desimal tidak didahului angka ataupun huruf lain
- Untuk bilangan biner didahului dengan &B
- Untuk bilangan heksadesimal didahului dengan &H

Contoh : 240 (bilangan desimal), &B11110000 (bilangan biner), &HF0 (bilangan heksadesimal)

2.7 Operator

Operator digunakan untuk melakukan operasi terhadap bilangan. Pada Bascom operator dibedakan menjadi operator aritmetik, operator relasional, dan operator logika. Operator aritmatik adalah operator yang digunakan dalam kalkulasi, yaitu + (penjumlahan), - (pengurangan), * (perkalian), / (pembagian), \ (pembagian integer), MOD (modulo = sisa dari pembagian). Operator relasional digunakan untuk membandingkan dua nilai, yang memberikan hasil benar (1) atau salah (0) dan dapat digunakan untuk membuat keputusan.

Tabel 2-3 Macam operator relasi pada Bascom

Operator	Relasi	Pernyataan
=	Sama dengan	X=Y
<>	Tidak sama dengan	X<>Y
<	Lebih kecil	X<Y
>	Lebih besar	X>Y
<=	Lebih kecil atau sama dengan	X<=Y
>=	Lebih besar atau sama dengan	X>=Y

Operator logika digunakan untuk menguji suatu pola bit tertentu, manipulasi bit atau operator Boolean. Misal operator AND dapat digunakan untuk mengabaikan semua bit dalam suatu byte kecuali satu bit untuk memantau status bit tersebut

Tabel 2-4 Macam operator logika pada Bascom

Operator	Makna
NOT	Komplemen/inverter
AND	Konjungsi (dan)
OR	Disjungsi (atau)
EXOR	EXclusive OR

2.8 Pernyataan Bersyarat

Pada Bascom terdapat beberapa pernyataan bersyarat yang sering digunakan yaitu If – Then, If – Then – Elseif, dan Select – Case

Sintaksis If – Then

```
If <syarat> Then
<Pernyataan 1>
<Pernyataan 2>
Else
<Pernyataan 3>
<Pernyataan 4>
End If
```

Sintaksis If – Then – Elseif

```
If <syarat1> Then
<Pernyataan 1>
<Pernyataan 2>
Elseif <syarat2>Then
<Pernyataan 3>
<Pernyataan 4>
Else
<Pernyataan 5>
<Pernyataan 6>
End If
```

Sintaksis Select – Case

```
Select Case < Variabel>
Case < Nilai 1> : <Pernyataan 1>
Case < Nilai 2> : <Pernyataan 2>
Case Else : <Pernyataan 3>
End Select
```

2.9 Pernyataan Perulangan (Loop)

Loop adalah suatu perulangan terhadap perintah atau instruksi sampai mencapai keadaan tertentu (jumlah perulangan tersebut dapat diketahui). Fungsi dari loop sendiri banyak sekali, dan dapat menghemat dalam penulisan program karena program yang sama dapat dilakukan dengan beberapa perintah dan kemudian diulang-ulang. Terdapat 3 pernyataan perulangan yaitu Do-Loop, While-Wend, dan For-Next.

Sintaksis Do – Loop

```
Do
<Pernyataan 1>
<Pernyataan 2>
Loop
```

14 Belajar Pemrograman Mikrokontroler AT89S51 Menggunakan Bascom-8051

Sintaksis While – Wend

```
While <Syarat>  
<Pernyataan 1>  
<Pernyataan 2>  
Wend
```

Sintaksis For – Next

```
FOR <Variabel> = <Nilai Awal> TO/DOWNTO <Nilai Akhir>  
<Pertambahan/Pengurangan>  
<Pernyataan 1>  
<Pernyataan 2>  
Exit
```

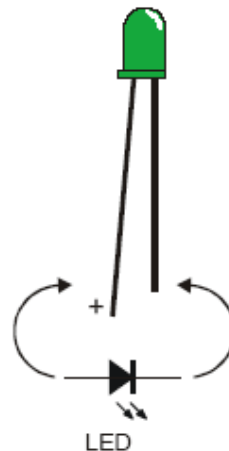

3. Output LED

Tujuan : Setelah melaksanakan modul ini mahasiswa diharapkan dapat :

- Membedakan dasar-dasar output pada mikrokontroler
- Memanfaatkan bilangan biner dan heksadesimal dalam pemrograman mikrokontroler
- Membuat rangkaian output sekaligus membuat program yang dapat mengendalikan konfigurasi nyala LED

LED (Light Emitting Diode) adalah semikonduktor (diode) yang dapat mengeluarkan cahaya. Terdapat berbagai macam warna LED, yaitu merah, hijau, oranye, kuning, dan biru, serta dalam berbagai bentuk. Seperti juga diode, LED juga merupakan komponen yang akan aktif (menyala) jika dialiri arus pada arah tertentu, dan tidak untuk arah arus sebaliknya. Gambar 3-1 menunjukkan bentuk fisik LED beserta simbolnya. Kaki yang lebih panjang adalah anode, yang akan dihubungkan dengan kutub +, dan kaki yang lebih pendek adalah katode, yang akan dihubungkan dengan kutub - .

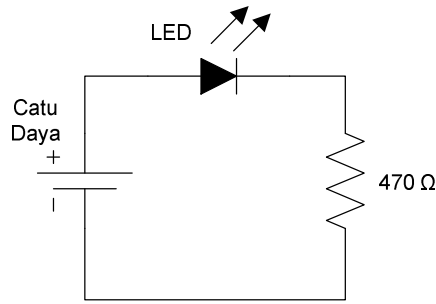
➤ Ingat KNAP : Katode (ke) Negatif, Anode (ke) Positif.



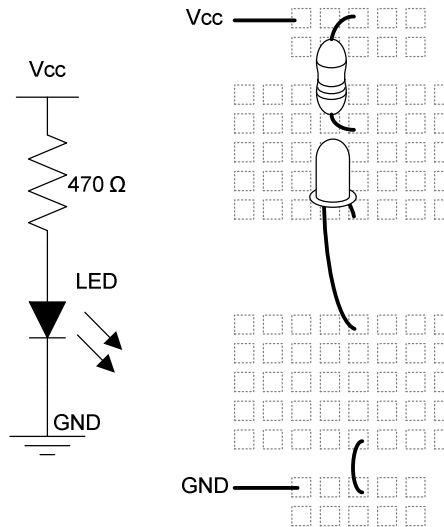
Gambar 3-1 Komponen LED beserta simbolnya

Gambar 3-2 menunjukkan rangkaian untuk menyalakan LED, anode terhubung dengan kutub positif catu daya dan katode terhubung dengan resistor dan ground. Fungsi resistor pada rangkaian yang melibatkan LED adalah sebagai pembatas arus. Untuk menyalakan LED hanya diperlukan arus sebesar 1 – 10 mA, sedangkan arus yang dapat diberikan catu daya (dengan tegangan berkisar 5-9V) cukup besar. Jika arus yang masuk LED tidak dibatasi, maka LED akan rusak. Resistor dapat diletakkan pada sisi katode maupun anode dari LED.

➤ Pada semua rangkaian LED jangan lupa untuk selalu diberikan resistor sebagai pembatas arus



Gambar 3-2 Rangkaian untuk menyalakan LED

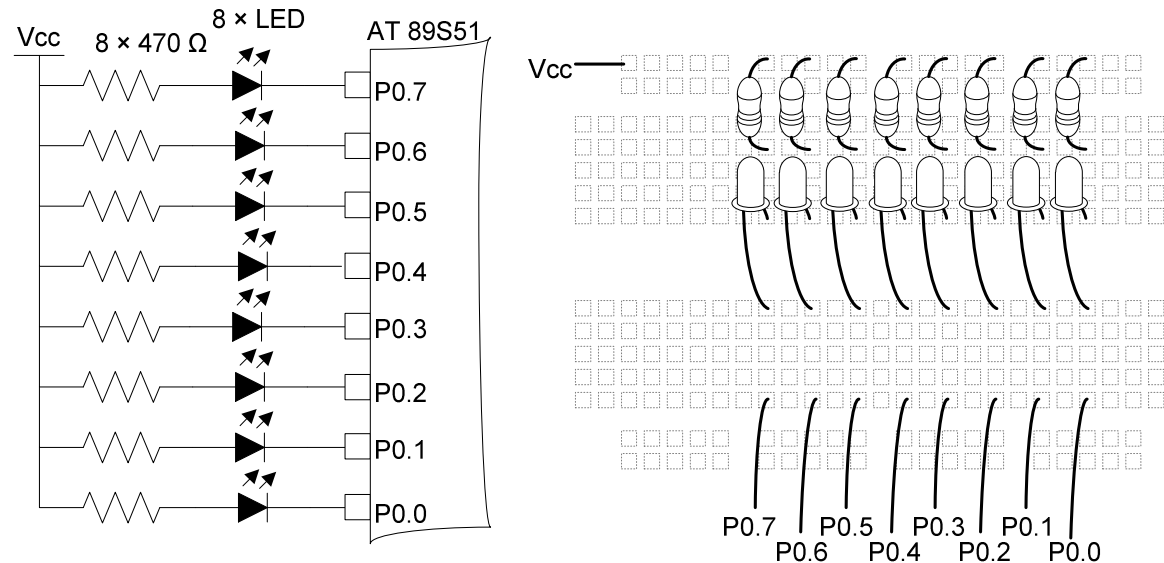


Gambar 3-3 Rangkaian dasar LED dan diagram pengkabelannya

3.1 Menyalakan dan Memadamkan LED

3.1.1 Rangkaian

Untuk mencoba rangkaian menyalakan/memadamkan LED, kita akan membuat rangkaian yang terdiri dari 8 buah LED, yang akan dikendalikan oleh mikrokontroler melalui port 0.



Gambar 3-4 Rangkaian Output LED

Program 3-1 Program LED_1.bas

```

'=====
'Program LED_1.bas
'Program untuk menyalakan Led pada P0.0, P0.1, P0.2, P0.3
'dan memadamkan LED pada P0.4, P0.5, P0.6, P0.7
'=====

P0.0 = 0          'P0.0, P0.1, P0.2, P0.3 diberi
logika 0          'sehingga LED nyala
P0.1 = 0
P0.2 = 0
P0.3 = 0
P0.4 = 1          'P0.4, P0.5, P0.6, P0.7 diberi
logika 1          'sehingga LED padam
P0.5 = 1
P0.6 = 1
P0.7 = 1

End               'akhir program

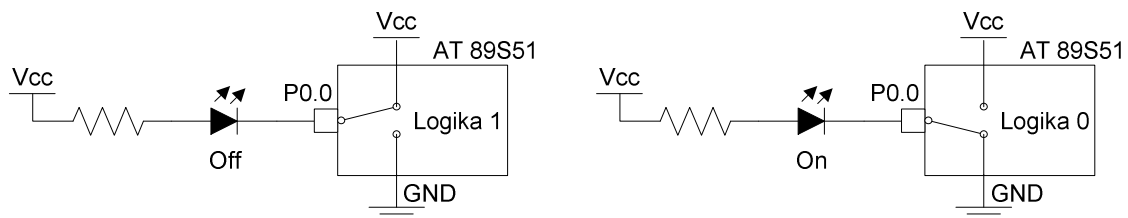
```

3.1.2 Prinsip Kerja Program LED_1.bas

Pada program ini port 0 berfungsi sebagai output, dengan masing-masing pin P0.0, P0.1, P0.2, dan P0.3 diberi logika 0 dan pin P0.4, P0.5, P0.6, dan P0.7 diberikan logika 1.

Pada dasarnya prinsip kerja output mikrokontroler adalah mirip saklar yang mempunyai dua kondisi, yaitu :

- menghubungkan rangkaian dengan Vcc jika diberi logika 1
- menghubungkan rangkaian dengan GND jika diberi logika 0.



Gambar 3-5 Logika 0 dan 1 pada output mikrokontroler

Gambar 3-5(a) menunjukkan saat P0.0 diberi logika 1, maka rangkaian terhubung dengan Vcc, akibatnya tidak ada beda tegangan pada rangkaian (sama-sama Vcc) sehingga tidak ada arus mengalir dan LED akan padam. Gambar 3-5(b) menunjukkan saat P0.0 diberi logika 0, maka rangkaian terhubung dengan GND sehingga terjadi beda tegangan pada rangkaian. Karena anode LED terhubung dengan Vcc dan katode LED terhubung dengan GND, maka arus akan mengalir pada LED dan LED akan menyala. Rangkaian pada Gambar 3-4 disebut sebagai rangkaian active low, yaitu rangkaian akan aktif jika diberi logika 0 (rendah) dan tidak aktif jika diberi logika 1 (tinggi).

Untuk memberikan logika 0 atau 1 kepada suatu pin dapat dilakukan secara individual per masing-masing port, seperti terdapat pada program LED_1.bas, ataupun secara sekaligus 1 port (yang terdiri dari 8 pin). Untuk memberikan logika secara sekaligus dipilih di antara tiga sistem bilangan, yaitu biner, heksadesimal atau desimal. Misal untuk konfigurasi nyala/padam seperti pada LED_1.bas, yaitu P0.0 hingga P0.3 menyala dan P0.4 hingga P0.7 padam, dapat dipandang sebagai bilangan biner 11110000b (Tabel 3-1).

Tabel 3-1 Keadaan masing-masing pin dan bilangan biner ekivalennya

P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	Biner
1	1	1	1	0	0	0	0	= 11110000b

Bilangan 11110000b jika dikonversi menjadi bilangan heksadesimal adalah F0h atau 240 desimal.

➤ Pelajari lagi tentang sistem bilangan. Anda juga dapat memanfaatkan program Calculator yang terdapat pada folder Accessories yang terdapat pada sistem operasi Windows untuk mengkonversi sistem bilangan.

Cara penulisan bilangan pada Bascom adalah :

- Untuk bilangan desimal tidak didahului angka ataupun huruf lain
- Untuk bilangan biner didahului dengan &b
- Untuk bilangan heksadesimal didahului dengan &h
- Contoh : 240 (bilangan desimal), &B11110000 (bilangan biner), &HF0 (bilangan heksadesimal)

Program LED_1.bas dapat dituliskan menjadi lebih ringkas seperti terdapat pada program led_2.bas

Program 3-2 Program LED_2.bas

```
'=====
'Program LED_2.bas
'sama seperti program LED_1, hanya berbeda cara pengisian,
'port P0 diberi nilai secara sekaligus
'=====

P0 = &HF0                                'memberikan nilai &HF0 ke P0
                                           'atau dapat dituliskan juga
                                           'P0 = &b11110000 atau
                                           'P0 = 240
```

End

Latihan 1

1. Buat program untuk mengendalikan LED nyala/padam dalam konfigurasi berikut :
 - a. P0.0 on, P0.1 off, P0.2 on, P0.3 off, P0.4 on, P0.5 off, P0.6 on, P0.7 off
 - b. P0.0 off, P0.1 on, P0.2 off, P0.3 on, P0.4 off, P0.5 on, P0.6 off, P0.7 on
 - c. P0.0 off, P0.1 off, P0.2 on, P0.3 ofn, P0.4 ffn, P0.5 on, P0.6 on, P0.7 off
 Masing-masing konfigurasi dibuat dengan dua cara, yaitu secara individual (lihat Program LED_1.bas) dan sekaligus (lihat program led_2.bas).

3.2 LED Berkedip

Untuk kegiatan ini rangkaian masih sama seperti 3.1 yaitu menggunakan gambar 3-4.

Program 3-3 Program LED_3.bas

```
'=====
'Program LED_3.bas
'Program untuk menyalakan dan mematikan LED pada P0 secara bergantian
```

```

=====
Do                                'awal do-loop
P0 = 0                            'semua LED menyala
Wait 1                            'tunggu 1 detik
P0 = &HFF                         'semua LED padam
Wait 1                            'tunggu 1 detik
Loop                               'akhir do-loop

End                                'akhir program

```

3.2.1 Prinsip kerja program LED_3.bas

Program ini adalah program untuk membuat LED berkedip dalam dua konfigurasi secara bergantian, yaitu nyala semua (P0 = 0), tundaan waktu selama 1 detik, padam semua (P0=&HFF), tundaan waktu 1 detik, dan kembali ke nyala semua. Pernyataan Do.... Loop digunakan agar program berjalan secara terus menerus

Pernyataan Wait 1 digunakan untuk memberikan waktu tunda selama 1 detik, jika diperlukan waktu tunda yang lebih singkat, dapat digunakan pernyataan Waitms. Misal pernyataan Waitms 500 akan memberikan waktu tunda selama 500 milidetik atau 0,5 detik.

➤ 1 detik = 1000 milidetik = 1000000 mikrodetik

Jika konfigurasi nyala LED yang akan dibuat cukup banyak, maka dapat dipergunakan pernyataan Lookup untuk mengambil data dari suatu tabel.

Program 3-4 Program LED_4.bas

```

=====
'Program LED_4.bas
'Program untuk membuat beberapa konfigurasi nyala LED
'menggunakan pernyataan Lookup
'=====
Dim N As Byte                    'deklarasi variabel

Do

For N = 0 To 3                  'awal For-Next
P0 = Lookup(N , Tabel)         'menampilkan data ke-N pada Tabel
ke P0
Wait 1                          'tunggu 1 detik
Next                            'akhir For-Next

Loop

End

Tabel:
Data 0 , &H0F , &HF0 , &HFF      'data pada Tabel

```

3.2.2 Prinsip Kerja Program LED_4.bas

Program LED_4.bas diawali dengan deklarasi variabel N sebagai Byte. Setiap variabel yang terdapat di program haruslah dideklarasikan terlebih dahulu menjadi salah satu tipe data.

➤ Tipe data pada Bascom ada beberapa macam, yang sering ditemui adalah Bit (bernilai 0 atau 1), Byte (bernilai antara 0 hingga 255), Integer (antara -32768 hingga 32767), Word (0 hingga 65535)

Pada program ini terdapat 2 loop, yaitu menggunakan pernyataan For...Next dan Do...Loop. Pernyataan di antara For N = 0 To 3 dan Next akan dilaksanakan 4 kali, dengan nilai N akan selalu bertambah 1 pada tiap loopnya. Nilai awal N adalah 0 dan nilai akhirnya adalah 4.

Pernyataan Lookup(N, Tabel) akan mengambil data indeks ke-N pada label Tabel. Indeks data dimulai dengan 0. Misal untuk N = 0, maka data yang diambil adalah 0, data ke-1 adalah &H0F, data ke-2 adalah &HF0, dan data ke-3 adalah &HFF.

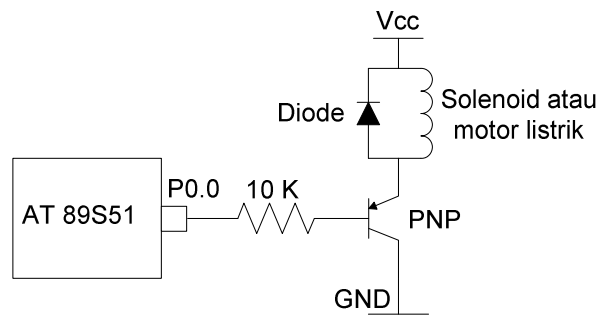
Jika program dijalankan akan menghasilkan 4 konfigurasi nyala LED secara bergantian dan dilakukan secara terus menerus.

Latihan 2

1. Buat program untuk membuat LED berkedip secara terus menerus dengan urutan 11111110b, 11111100b, 11111001b, 11110011b, 11100111b, 11001111b, 10011111b, 00111111b, 01111110b
2. Seperti pada latihan sebelumnya, tapi kali ini LED hanya melakukan pergeseran sebanyak 5 kali, kemudian semua LED menyala.

Diskusi 1

1. Selain rangkaian LED active low, terdapat juga rangkaian LED active high. Diskusikan dengan teman sekelompok anda prinsip kerja rangkaian LED active high. Untuk mikrokontroler AT89S51 mengapa lebih dipilih rangkaian active low. (Petunjuk : lihat data sheet AT89S51).
2. Gambar 3-6 menunjukkan contoh rangkaian untuk mengendalikan solenoid atau motor listrik. Diskusikan prinsip kerjanya. Apakah rangkaian ini termasuk active high atau active low? Apa fungsi diode pada rangkaian ini? Jika transistor PNP diganti dengan NPN perubahan apa yang diperoleh?



Gambar 3-6 Rangkaian output untuk mengendalikan solenoid atau motor listrik

4. Input Saklar

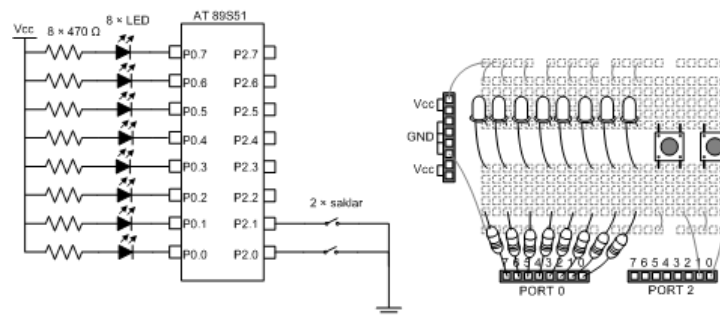
Tujuan : Setelah melaksanakan modul ini mahasiswa diharapkan dapat :

- Membuat rangkaian input sekaligus membuat program yang dapat menunjukkan kondisi saklar

Saklar adalah peranti input digital yang umum ditemui dalam banyak peralatan. Fungsi kerja saklar adalah untuk menghubungkan atau memutus aliran listrik pada suatu rangkaian listrik. Pada saklar terdapat 2 kemungkinan keadaan (biner), yaitu aktif dan tidak aktif. Jika suatu saklar menyambung rangkaian listrik saat aktif (ditekan) disebut sebagai saklar Normally Open (NO), jika sebaliknya disebut sebagai Normally Closed (NC). Berdasarkan cara pengaktifannya, saklar dapat dibedakan menjadi saklar tekan (pushbutton) dan saklar toggle/slide. Saklar tekan akan langsung kembali tidak aktif jika tidak ditekan, sedangkan saklar toggle baru akan tidak aktif jika posisi saklar dipindahkan.

4.1 Menampilkan Kondisi Saklar Pada LED

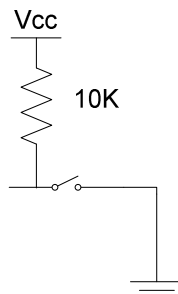
4.1.1 Rangkaian



Gambar 4-1 Rangkaian saklar dan LED

Rangkaian ini menggunakan rangkaian saklar active low, dimana jika saklar ditekan (aktif) maka oleh mikrokontroler dibaca 0. Sebaliknya jika saklar tidak diaktifkan mikrokontroler akan membacanya sebagai logika 1, walaupun sebetulnya keadaan rangkaian adalah floating (mengambang), tidak terhubung dengan Vcc ataupun GND. Hal ini disebabkan port input/output AT89S51 kompatibel dengan input/output dari TTL.

- TTL adalah salah satu jenis IC logika selain CMOS. Rangkaian seperti ini tidak dapat digunakan untuk semua jenis mikrokontroler, lebih disarankan untuk menggunakan resistor pull up. Resistor pull up digunakan untuk memastikan saat saklar tidak aktif, rangkaian tetap terhubung dengan Vcc. Penjelasan lebih lanjut dapat diperoleh pada buku tentang rangkaian logika.



Gambar 4-2 Rangkaian input saklar menggunakan resistor pull up

Program 4-1 Program Saklar_1.bas

```
'=====
'Program Saklar_1.bas
'Program untuk menampilkan kondisi saklar pada LED
'=====

Saklar Alias P2          'P2   digantikan   dengan   nama
variabel Saklar
Led Alias P0            'P0   digantikan   dengan   nama
variabel LED

Do

Led = Saklar            'data dari Saklar dikirim ke LED

Loop

End
```

4.1.2 Prinsip Kerja Program Saklar_1.bas

Program ini memindahkan data dari P2 yang terhubung dengan input saklar ke P0 yang terhubung dengan output LED. Pada awal program terdapat pernyataan Alias yang dipergunakan untuk menggantikan penyebutan P2 dan P0 masing-masing dengan nama variabel Saklar dan LED. Hal ini akan berguna pada saat membuat program yang panjang dan kompleks, jika terdapat perubahan penggunaan pin atau port, cukup diganti pada pernyataan Alias.

4.2 Saklar Untuk Memilih Konfigurasi Nyala LED

Program 4-2 Program Saklar_2.bas

```
'=====
'Program Saklar_1.bas
'Program untuk menampilkan kondisi saklar pada LED
'=====

Saklar Alias P2
Led Alias P0

Do

    If Saklar = &HFF Then
        Led = &HFF
    ElseIf Saklar = &HFE Then
        Led = &HAA
    Else
        Led = 0
    End If

Loop

End
```


4.2.1 Prinsip Kerja Program Saklar_2.c

Pada program ini kondisi saklar akan menentukan konfigurasi nyala LED. Jika tidak ada saklar yang diaktifkan maka semua LED akan padam, jika hanya saklar P2.0 saja yang diaktifkan maka LED akan membentuk konfigurasi nyala/padam secara berselang seling. Jika selain dari dua keadaan tersebut maka LED akan menyala semua.

Untuk menguji keadaan saklar digunakan pernyataan If...Elseif...Else...Endif, sehingga jika keadaan saklar sesuai dengan keadaan yang disyaratkan, maka pernyataan setelah Then akan dilaksanakan.

Latihan 3

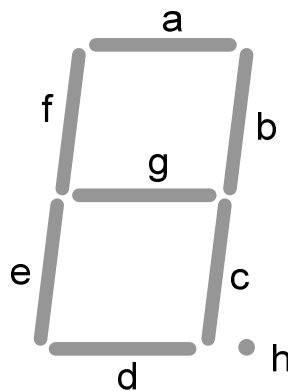
1. Buat program dengan kondisi sebagai berikut. Jika tidak ada saklar diaktifkan maka semua LED akan padam, jika hanya saklar P2.0 yang diaktifkan maka semua LED akan bergantian menyala dan padam.

5. Display 7 Segment

Tujuan : Setelah melaksanakan modul ini mahasiswa diharapkan dapat :

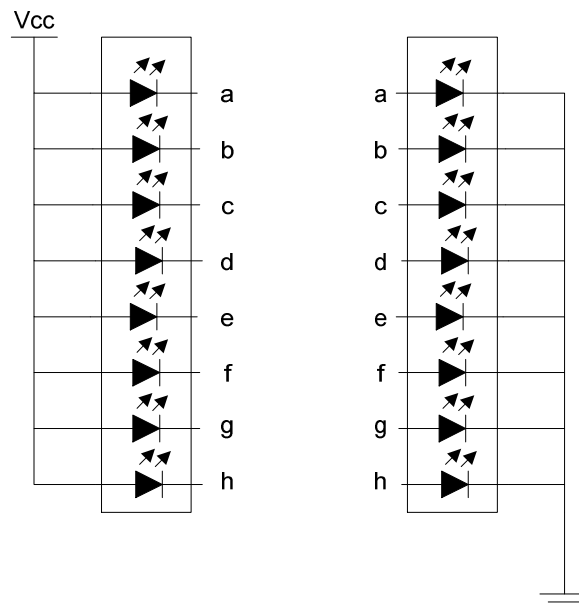
- Membedakan 7 segment jenis common anode dan common cathode
- Membuat rangkaian output sekaligus membuat program yang dapat mengendalikan tampilan 7 segment

7 segment adalah salah satu komponen yang paling banyak dipergunakan untuk menampilkan karakter berupa angka. Pada dasarnya 7 segment adalah rangkaian 7 buah LED dijadikan satu. Untuk dapat membentuk suatu karakter maka LED-LED tersebut diatur nyala/padamnya secara terpisah. Pada suatu 7 segment juga ditambahkan satu LED untuk dot sebagai penunjuk desimal. Untuk memudahkan dalam mengidentifikasi, masing-masing segmen yang berupa LED diberi notasi a, b, c, d, e, f, g dan h (gambar 5-1).



Gambar 5-1 Display 7 segment beserta notasi masing-masing segmen

Untuk menyederhanakan rangkaian, maka pada 7 segment salah satu dari anode atau katode digabungkan dalam satu pin. Jika anode yang digabungkan disebut sebagai common anode, sedangkan jika katode yang digabungkan disebut sebagai common cathode.



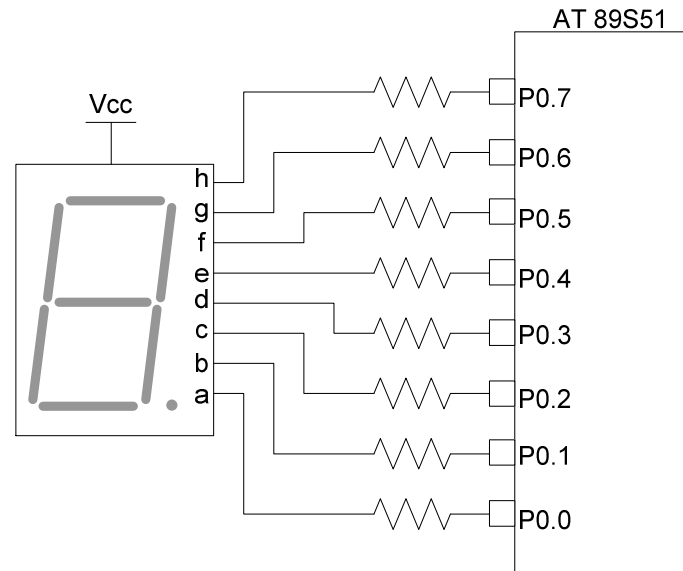
Gambar 5-2 Jenis-jenis 7 segment (a) common anode (b) common cathode

26 Belajar Pemrograman Mikrokontroler AT89S51 Menggunakan Bascom-8051

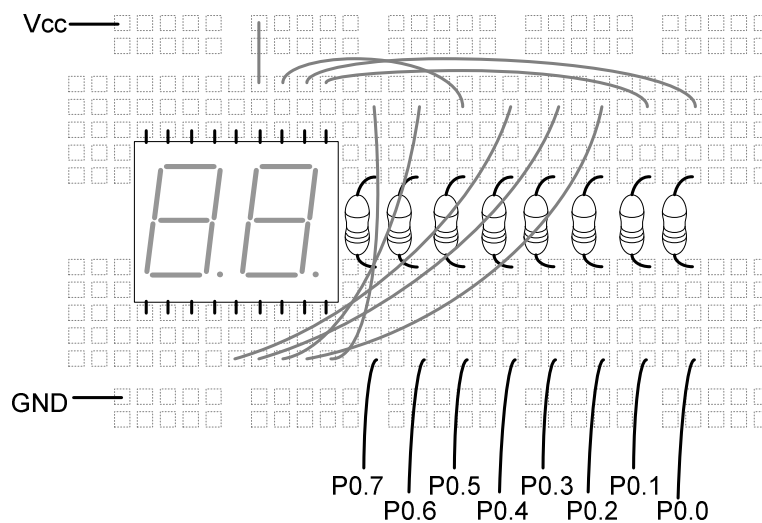
Gambar 5-2 menunjukkan skema dari kedua jenis 7 segment tersebut. Untuk jenis common anode, segmen akan menyala jika pin dihubungkan dengan GND, jika dihubungkan dengan Vcc maka segmen akan padam. Sebaliknya pada jenis common anode, segmen akan menyala jika dihubungkan dengan Vcc dan padam jika dihubungkan dengan GND.

5.1 Kegiatan 1: Menampilkan Karakter Angka Pada 7 Segment Tunggal

5.1.1 Rangkaian



Gambar 5-3 Rangkaian display menggunakan 7 segment CA



Gambar 5-4 Diagram pengkabelan display 7 segment CA

Program 5-1 Program 7seg_1.bas

```
'=====
'Program 7seg_1.bas
'Program untuk menampilkan karakter angka 0 pada 7 segment
```

```
'=====
=====
P0 = &HC0
End
```

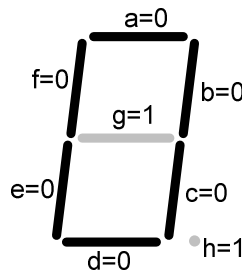
5.1.2 Prinsip Kerja Program 7seg_1.bas

Program ini akan menampilkan karakter angka 0 ke 7 segment dengan mengirimkan data &HC0 ke P0. Rincian logika masing-masing pin P0.0 hingga P0.7 dapat dilihat pada tabel 5-1.

Tabel 5-1 Data yang dikirim ke 7 segment untuk menampilkan karakter 0

heksa	biner	P0.7=h	P0.6=g	P0.5=f	P0.4=e	P0.3=d	P0.2=c	P0.1=b	P0.0=a
&HC0	11000000b	1	1	0	0	0	0	0	0

Berdasarkan prinsip kerja 7 segment CA, maka segmen yang diberikan logika 0 (dihubungkan dengan Vcc) akan menyala, sedangkan segmen yang diberikan logika 1 (dihubungkan dengan GND) akan padam, maka segmen a, b, c, d, e, dan f akan menyala, sedangkan segmen g dan h akan padam. Hasilnya adalah akan terbentuk karakter 0 pada 7 segment seperti terlihat pada gambar 5-5.



Gambar 5-5 7 segment menampilkan karakter angka 0

Diskusi 2

1. Lengkapi tabel 5-2 di bawah ini dengan data yang diberikan ke P0 untuk menampilkan karakter angka 1 hingga 9.

Tabel 5-2 Data yang dikirim ke 7 segment untuk menampilkan karakter 0 – 7

Karakter	P0.7 = h	P0.6 = g	P0.5 = f	P0.4 = e	P0.3 = d	P0.2 = c	P0.1 = b	P0.0 = a	biner	heksa
0	1	1	0	0	0	0	0	0	11000000b	&hc0
1										
2										
3										
4										
5										
6										
7										
8										
9										

Latihan 4

1. Buat program untuk menampilkan karakter angka 0 dan 1 secara bergantian dengan memodifikasi program LED_2a.c

28 Belajar Pemrograman Mikrokontroler AT89S51 Menggunakan Bascom-8051

2. Buat program untuk menampilkan karakter angka 0 diurutkan naik hingga 9 secara bergantian dengan memodifikasi program LED_2b.c
3. Dengan memasang modul input saklar ke P2, buat program untuk menentukan karakter yang ditampilkan pada 7 segment sesuai dengan saklar yang diaktifkan. Jika saklar P2.0 aktif maka akan ditampilkan karakter angka 0, jika saklar P2.1 diaktifkan maka akan ditampilkan karakter angka 1, dan seterusnya hingga P2.8 akan menampilkan karakter angka 8.
4. Buat program untuk menampilkan karakter angka 0 diurutkan naik hingga 9 jika yang diaktifkan hanya P2.0, sedangkan jika hanya P2.1 yang diaktifkan maka karakter angka akan diurutkan turun. Selain kedua kondisi saklar tersebut maka 7 segment akan padam.

6. Display 7 Segment Menggunakan Rangkaian Multipleks

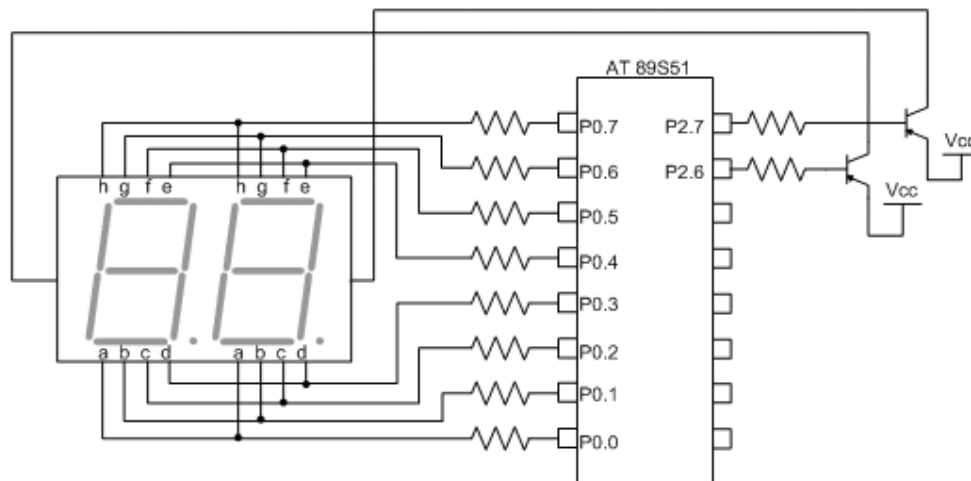
Tujuan : Setelah melaksanakan modul ini mahasiswa diharapkan dapat :

- Membuat rangkaian output sekaligus membuat program untuk 7 segment yang menampilkan lebih dari satu digit

Untuk menghemat output saat mengendalikan 7 segment dengan jumlah 2 atau lebih, dipergunakan cara multiplekser. Cara ini memanfaatkan “kelemahan” mata manusia yang tidak dapat melihat kedipan yang sangat cepat. Sebetulnya sinyal yang diberikan kepada masing-masing 7 segment tidak dilakukan secara bersamaan tetapi bergantian, namun karena dilakukan secara cepat kita tetap melihat bahwa kedua 7 segment menyala secara bersamaan.

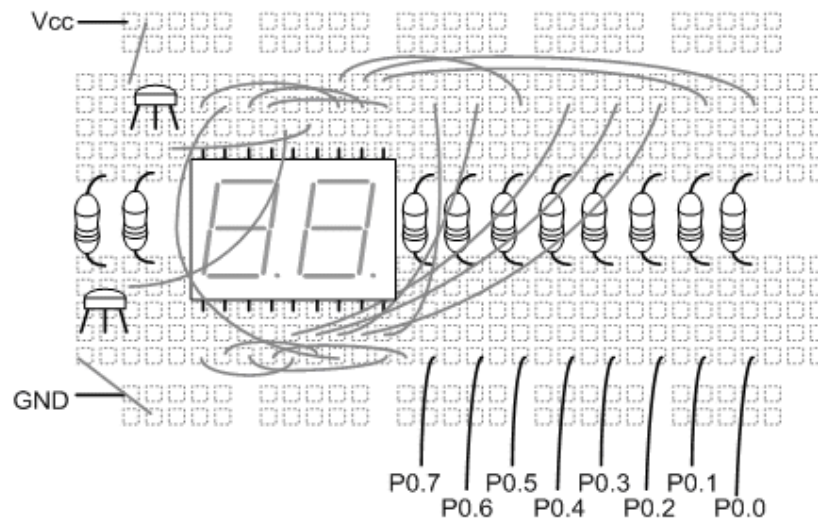
6.1 Menampilkan Karakter Angka 0 – 99 Pada Display 7 Segment Multipleks

6.1.1 Rangkaian



Gambar 6-1 Rangkaian multiplekser 2 buah 7 segment

Kedua 7 segment akan dinyalakan bergantian dengan cara memberikan logika 0 melalui P1.0 untuk menyalakan digit satuan. Sedangkan jika P1.1 yang diberikan logika 0 maka digit puluhan yang dinyalakan.



Gambar 6-2 Diagram pengkabelan rangkaian multiplexer 2 buah 7 segment

Program 6-1 Program 7seg_mux.bas

```

=====
'Program 7seg_mux.bas
'Program untuk menampilkan karakter angka 0 -99
'pada rangkaian multiplexer 7 segment
=====
Dim Angka As Word , Sisa As Byte
Dim Puluhan As Byte , Satuan As Byte
Dim N As Byte

Do
For Angka = 0 To 99 Step 1      'untuk incremental 1
                                'pernyataan "step 1" boleh
                                dihilangkan
    Gosub Disp_7s                'panggil subrutin Disp_7s
Next
Loop

End

Disp_7s:                        'subrutin untuk menampilkan 2
karakter                        'pada 7 segmen
                                'sisa pembagian Angka dengan 100

Sisa = Angka Mod 100
Puluhan = Sisa / 10
Satuan = Sisa Mod 10
For N = 0 To 20
    P0 = &HFF
    P1 = &HFE
    P0 = Lookup(satuan , 7_seg)
    Waitms 10
    P0 = &HFF
    P1 = &HFD
    P0 = Lookup(puluhan , 7_seg)
    Waitms 10
Next
Return                            'kembali ke program utama

```



```

7_seg:          'tabel lookup untuk kode 7
segment
Data &HC0 , &HF9 , &HA4 , &HB0 , &H99
Data &H92 , &H82 , &HF8 , &H80 , &H90

```

6.1.2 Prinsip Kerja Program 7seg_mux.bas

Program ini menampilkan karakter angka secara berurutan dari 0 hingga 99 yang terdapat pada pernyataan For Angka = 0 To 99. Angka tersebut ditampilkan dengan memanggil subrutin Disp_7s.

Berbeda dengan pemahaman kita, bahwa angka 40 merupakan satu kesatuan, untuk menampilkannya pada 7 segment maka digit 1 (puluhan) dan digit 2 (satuan) akan diberikan karakter secara terpisah, yaitu 4 ke digit 1 dan 0 ke digit 0. Subrutin Disp_7s digunakan untuk memisahkan digit yang akan dikirimkan ke 7 segment kemudian menyalakan 7 segment sesuai dengan data yang bersesuaian (data karakter 4 dikirim saat digit puluhan menyala, data karakter 2 dikirim saat digit satuan menyala), lalu diberikan efek penundaan. Pemisahan digit satuan dan puluhan terdapat pada pernyataan

```

Puluhan = Sisa / 10
Satuan = Sisa Mod 10

```

Pernyataan

```
Sisa = Angka Mod 100
```

digunakan untuk mengantisipasi jika nilai variabel Angka lebih dari 100, maka yang ditampilkan hanyalah digit satuan dan puluhan.

Pernyataan

```

P0 = &HFF
P1 = &HFE
P0 = Lookup(satuan , 7_seg)
Waitms 10

```

menunjukkan bahwa saat data satuan yang dikirimkan, 7 segment digit satuan akan dinyalakan (P1.0 diberi logika 0), sedangkan pernyataan

```

Waitms 10
P0 = &HFF
P1 = &HFD
P0 = Lookup(puluhan , 7_seg)
Waitms 10

```

menunjukkan bahwa data puluhan dikirimkan saat 7 segment digit puluhan dinyalakan (P1.1 diberi logika 0).

Latihan 5

Buat program untuk menampilkan karakter angka 0 diurutkan naik hingga 99 jika yang diaktifkan hanya P2.0, sedangkan jika hanya P2.1 yang diaktifkan maka karakter angka akan diurutkan turun. Selain kedua kondisi saklar tersebut maka 7 segment akan padam (pertimbangkan untuk menggunakan pernyataan increment dan decrement)

7. LCD

Tujuan : Setelah melaksanakan modul ini mahasiswa diharapkan dapat :

- Membuat rangkaian output sekaligus membuat program yang dapat mengendalikan tampilan LCD

LED dan 7 segment sangat umum ditemui pada berbagai macam peranti digital. Walau demikian untuk aplikasi yang membutuhkan informasi yang lebih kompleks diperlukan display yang dapat menampilkan lebih banyak karakter. Saat ini LCD (Liquid Crystal Display) semakin banyak dipergunakan sebagai pengganti rangkaian LED maupun 7 segmen. Hal ini disebabkan oleh

- semakin murah harga LCD
- kemampuan LCD untuk menampilkan berbagai macam karakter berupa angka dan huruf, bahkan grafik.
- Pemrograman LCD yang semakin mudah

LCD dipergunakan antara lain untuk menampilkan pesan atau informasi kepada pengguna (misal display microwave atau jam digital) dan untuk menampilkan menu input untuk perubahan setting (misal pada mesin fotokopi)

Terdapat banyak macam kontroler yang dipergunakan pada LCD, namun yang paling luas penggunaannya adalah Hitachi HD44780 dan kompatibelnya. LCD dibedakan berdasarkan jumlah baris dan kolom karakter yang ditampilkan. Misal LCD 2 x 16 menunjukkan 2 kolom dan 16 kolom karakter. Terdapat LCD dengan jumlah kolom baris 2,4, atau 8 dengan kolom karakter berjumlah 8, 16, 20, 24, 32, dan 40 kolom. Hampir semua LCD dapat diatur tingkat kontrasnya, sebagian mempunyai backlight. LCD yang kompatibel dengan HD44780 pada umumnya mempunyai 16 dengan fungsi seperti tertera pada tabel 7-1.

Tabel 7-1 Nama dan fungsi masing-masing pin pada LCD

Pin	Nama Pin	Input/Output	Fungsi
1	VSS	Power	Catu daya, ground
2	VDD	Power	Catu daya, 5 V
3	V0	Power	Tegangan referensi untuk pengaturan kontras
4	R/S	Input	Pemilihan register 0 = register instruksi 1 = register data
5	R/W	Input	Mode baca/tulis 0 = tulis ke LCD (write mode) 1 = baca dari LCD (read mode)
6	E	Input	Enable 0 = mulai latch data 1 = enable
7 – 14	DB0 – DB7	Input/Output	Data DB0 = LSB, DB7 = MSB Mode 8 bit digunakan DB0 – DB7 Mode 4 bit digunakan DB4 – DB7
15	BPL	Power	Catu daya backlight, positif
16	GND	Power	Catu daya backlight, negatif

34 Belajar Pemrograman Mikrokontroler AT89S51 Menggunakan Bascom-8051

Display karakter pada LCD diatur oleh pin EN, RS dan RW. Jalur E dinamakan Enable. Jalur ini digunakan untuk memberitahu LCD bahwa anda sedang mengirimkan sebuah data. Untuk mengirimkan data ke LCD, maka melalui program EN harus dibuat logika low "0" dan set pada dua jalur kontrol yang lain RS dan RW. Ketika dua jalur yang lain telah siap, set EN dengan logika "1" dan tunggu untuk sejumlah waktu tertentu (sesuai dengan datasheet dari LCD tersebut) dan berikutnya set EN ke logika low "0" lagi.

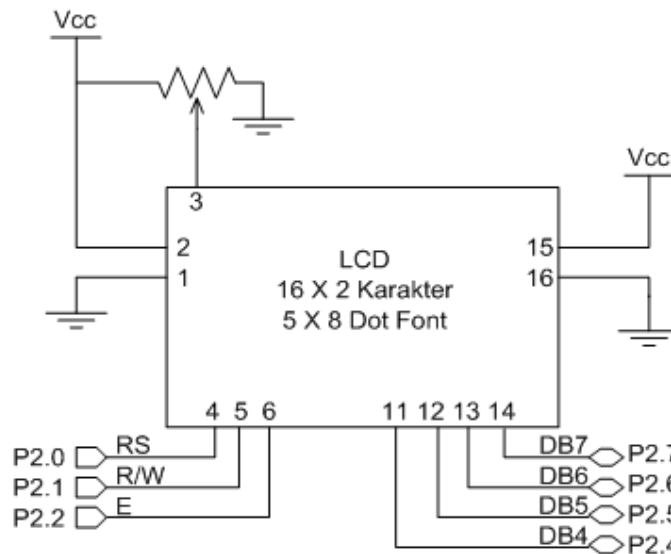
Jalur RS adalah jalur Register Select. Ketika RS berlogika low "0", data akan dianggap sebagai sebuah perintah atau instruksi khusus (seperti clear screen, posisi kursor dll). Ketika RS berlogika high "1", data yang dikirim adalah data text yang akan ditampilkan pada display LCD. Sebagai contoh, untuk menampilkan huruf "T" pada layar LCD maka RS harus diset logika high "1".

Jalur RW adalah jalur kontrol Read/ Write. Ketika RW berlogika low (0), maka informasi pada bus data akan dituliskan pada layar LCD. Ketika RW berlogika high "1", maka program akan melakukan pembacaan memori dari LCD. Sedangkan pada aplikasi umum pin RW selalu diberi logika low "0". Pada latihan ini LCD hanya akan dipergunakan pada mode tulis, sehingga pin R/W dihubungkan dengan ground

Terakhir, bus data terdiri dari 4 atau 8 jalur (bergantung pada mode operasi yang dipilih oleh user). Pada mode data 8 bit, jalur yang dipergunakan adalah DB0 – DB7, sedangkan untuk mode data 4 bit, jalur yang dipergunakan adalah DB4 – DB7.

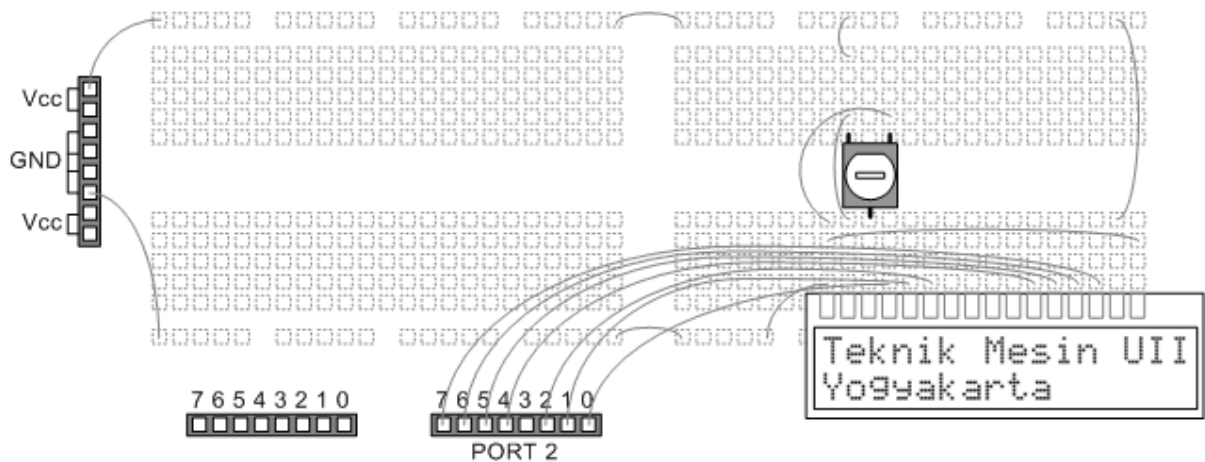
7.1 Kegiatan 1 : Menampilkan Pesan Sederhana Pada LCD

7.1.1 Rangkaian



Gambar 7-1 Koneksi LCD dengan mikrokontroler

Pada kegiatan ini mode data yang dipergunakan adalah mode data 4 bit, sehingga yang dipergunakan hanya DB4 – DB7. Jika mikrokontroler hanya dipergunakan untuk menuliskan karakter ke LCD maka pin R/W dapat dihubungkan dengan ground. Potensiometer dipergunakan untuk mengatur kontras karakter yang ditampilkan.



Gambar 7-2 Diagram pengkabelan koneksi LCD dengan mikrokontroler

Program 7-1 Program LCD_1.bas

```

'=====
'Program LCD_1.bas
'Program untuk menampilkan pesan teks sederhana pada LCD
'=====

'Inisialisasi LCD
Config Lcd = 16 * 2
Config Lcdpin = Pin , Db4 = P2.4 , Db5 = P2.5 , Db6 = P2.6 ,
Config Lcdpin = Pin , Db7 = P2.7 , E = P2.2 , Rs = P2.0
P2.1 = 0 'R/W=0
Initlcd

Cursor Off 'menonaktifkan cursor
Cls 'membersihkan layar
Upperline 'menampilkan pesan pada baris
pertama
Lcd "Teknik Mesin UII"
Lowerline 'menampilkan pesan pada baris
kedua
Lcd "Yogyakarta"

End

```

7.1.2 Prinsip Kerja Program LCD_1.bas

Semua program yang berkaitan dengan LCD diawali dengan inisialisasi LCD. Pernyataan

```
Config Lcd = 16 * 2
```

menentukan jumlah karakter LCD yang dipakai (16 kolom × 2 baris), sedangkan

```
Config Lcdpin = Pin , Db4 = P2.4 , Db5 = P2.5 , Db6 = P2.6 ,
Config Lcdpin = Pin , Db7 = P2.7 , E = P2.2 , Rs = P2.0
P2.1 = 0
'R/W=0
```

menentukan koneksi pin LCD dengan mikrokontroler

Untuk menonaktifkan kursor dan membersihkan layar digunakan pernyataan

```
Cursor Off 'menonaktifkan cursor
Cls 'membersihkan layar
```

36 Belajar Pemrograman Mikrokontroler AT89S51 Menggunakan Bascom-8051

Untuk pengaturan kursor terdapat beberapa pilihan, yaitu

```
Cursor Off                'kursor tidak aktif
Cursor On Blink           'kursor tidak aktif
Cursor Off Noblink       'kursor tidak aktif
```

Selanjutnya dituliskan teks yang diinginkan dengan menggunakan pernyataan

```
Upperline                'menampilkan pesan pada baris
pertama
Lcd "Teknik Mesin UII"
Lowerline                'menampilkan pesan pada baris
kedua
Lcd "Yogyakarta"
```

Latihan 6

1. Buat program seperti Program 7-1, yaitu namun berkedip-kedip, 0.5 s menyala, 0.5 s padam.
2. Buat program yang menampilkan karakter “Teknik Mesin UII” dan “Yogyakarta”, kemudian berganti menjadi “Laboratorium” dan “Mekatronika” kemudian kembali lagi ke tampilan.

Program 7-2 Program LCD_2.bas

```
'=====
=====
'Program LCD_2.bas
'Program untuk menampilkan variabel pada LCD
'=====
=====

Dim N As Byte

'Inisialisasi LCD
Config Lcd = 16 * 2
Config Lcdpin = Pin , Db4 = P2.4 , Db5 = P2.5 , Db6 = P2.6 ,
Config Lcdpin = Pin , Db7 = P2.7 , E = P2.2 , Rs = P2.0
P2.1                                =                                0
'R/W=0
Initlcd

Cursor Off                'menonaktifkan cursor
Cls                        'membersihkan layar

For N = 1 To 100
Home                      'meletakkan karakter di baris 1,
kolom 1
Lcd "Desimal"
Locate 1 , 14             'baris 1, kolom 14
Lcd N
Home Lowerline           'meletakkan karakter di baris 2,
kolom 1
Lcd "Heksadesimal"
Locate 2 , 14            'baris 2, kolom 14
Lcdhex N
Wait 1
Next

End
```

7.1.3 Prinsip Kerja Program LCD_2.bas

Sama seperti program LCD_1.bas program ini juga diawali dengan inisialisasi. Teks yang ditampilkan adalah “Desimal” dan angka desimal antara 1 hingga 100 pada baris pertama, serta “Heksadesimal” yang diikuti dengan angka heksadesimal.

Pernyataan

```
Home                               'meletakkan karakter di baris 1,
kolom 1
```

```
Lcd "Desimal"
```

```
Locate 1 , 14                       'baris 1, kolom 14
```

```
Lcd N
```

menunjukkan lokasi mulainya teks yaitu baris pertama dan kolom pertama LCD, kemudian teks “Desimal” dan angka desimal N yang diletakkan pada baris pertama dan kolom 14.

Pernyataan

```
Home Lowerline                     'meletakkan karakter di baris 2,
kolom 1
```

```
Lcd "Heksadesimal"
```

```
Locate 2 , 14                       'baris 2, kolom 14
```

```
Lcdhex N
```

menunjukkan mulainya teks pada baris kedua dan kolom pertama, menuliskan teks “Heksadesimal” dan bentuk heksadesimal dari N yang diletakkan pada baris kedua kolom 14.

Latihan 7

1. Tuliskan program Program 7-3 kemudian download dan jalankan. Jelaskan prinsip kerja program tersebut.

Program 7-3 Program LCD_timer.bas

```
'=====
'Program LCD_timer.bas
'Program untuk menampilkan penghitung waktu pada LCD
'=====

Dim Jam As Byte
Dim Menit As Byte
Dim Detik As Byte

'Inisialisasi LCD
Config Lcd = 16 * 2
Config Lcdpin = Pin , Db4 = P3.4 , Db5 = P3.5 , Db6 = P3.6,
Config Lcdpin = Pin , Db7 = P3.7 , E = P2.1 , Rs = P2.0
Initlcd

Cursor Off                               'menonaktifkan cursor
Cls                                       'membersihkan layar

Do
  If Detik = 60 Then
    Detik = 00
    Menit = Menit + 1
  End If

  If Menit = 60 Then
    Menit = 00
```

38 Belajar Pemrograman Mikrokontroler AT89S51 Menggunakan Bascom-8051

```
Jam = Jam + 1
End If

If Jam = 24 Then
    Jam = 00
End If

If Jam < 10 Then
    Lcd "0" ; Jam ; ":"
Else
    Lcd Jam ; ":"
End If

If Menit < 10 Then
    Lcd "0" ; Menit ; ":"
Else
    Lcd Menit ; ":"
End If

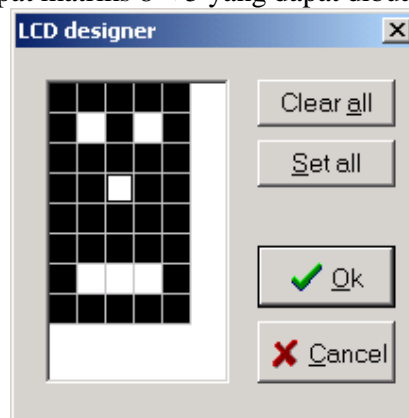
If Detik < 10 Then
    Lcd "0" ; Detik
Else
    Lcd Detik
End If

Waitms 10
Detik = Detik + 1
Cls
Loop

End
```

7.2 Kegiatan 2 : Menampilkan Karakter Buatan Sendiri Pada LCD

Pada Bascom terdapat fasilitas LCD designer untuk membuat sendiri karakter yang diinginkan dengan mengklik Tools – LCD designer. Gambar 7-3 menunjukkan pada jendela LCD designer terdapat matriks 8×5 yang dapat dibuat menjadi karakter tertentu.



Gambar 7-3 Jendela LCD designer

Setelah karakter selesai dibuat, kemudian pada jendela editor akan ditampilkan pernyataan seperti di bawah ini

```
Deflcdchar ?, 31, 21, 31, 27, 31, 31, 17, 31
```


Tanda ? harus digantikan dengan angka 0 – 7 yang akan digunakan sebagai nomor karakter untuk memanggil karakter. Misal karakter yang dipanggil adalah karakter dengan nomor karakter 0, maka dituliskan pernyataan

```
Lcd Chr(0)
```

Latihan 8

1. Tuliskan Program 7-3 kemudian download dan jalankan.

Program 7-4

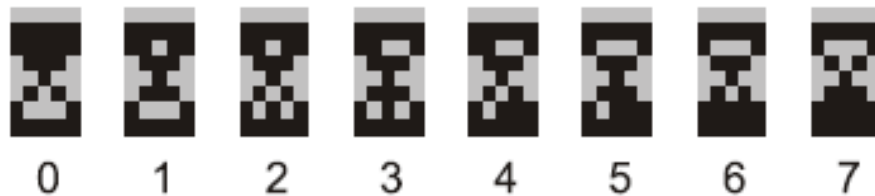
```
'=====
'Program chr_cust.bas
'Program untuk menampilkan variabel pada LCD
'=====

Dim N As Byte

'Inisialisasi LCD
Config Lcd = 16 * 2
Config Lcdpin = Pin , Db4 = P3.4 , Db5 = P3.5 , Db6 = P3.6 ,
Config Lcdpin = Pin , Db7 = P3.7 , E = P2.1 , Rs = P2.0
Initlcd

Deflcdchar 0 , 32 , 16 , 8 , 4 , 2 , 1 , 32 , 32
Deflcdchar 1 , 32 , 32 , 13 , 18 , 32 , 32 , 32 , 32
Cls
Lcd Chr(0)
Lcd Chr(1)
End
```

2. Buat program untuk menampilkan karakter di bawah ini secara bergantian, masing-masing 0.5 s.



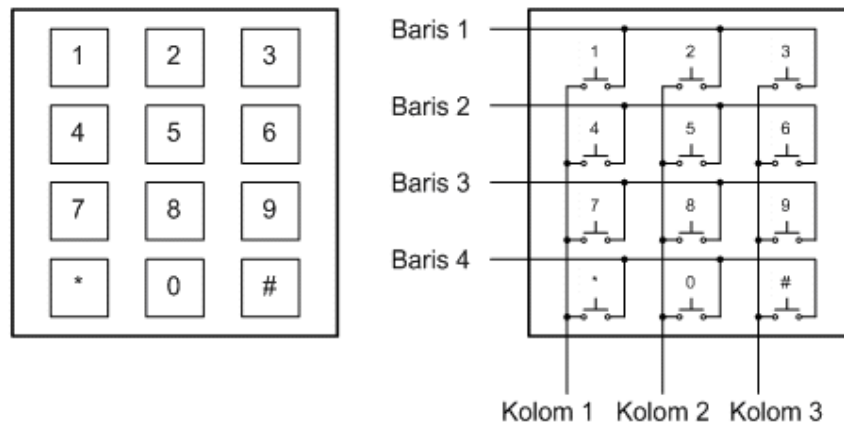
8. Keypad

Tujuan : Setelah melaksanakan modul ini mahasiswa diharapkan dapat :

- Membuat program yang dapat menampilkan tombol yang ditekan pada keypad

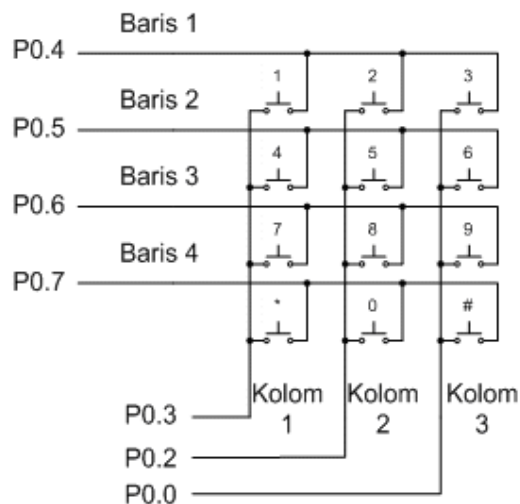
8.1 Membaca Tombol Keypad Yang Ditekan Untuk Ditampilkan Pada LED

Keypad adalah kumpulan saklar NO. Untuk menunjukkan posisi saklar, keypad dibagi menjadi baris dan kolom. Gambar menunjukkan contoh keypad 3 kolom 4 baris, yang lebih sering disebut sebagai keypad 3 x 4. Pin yang terhubung dengan baris akan bertindak sebagai output dari mikrokontroler, sedangkan pin kolom akan bertindak sebagai pin input. Untuk mendeteksi tombol mana yang ditekan, maka mikrokontroler akan mengirim sinyal low ke baris 1, kemudian mikrokontroler akan mengecek kolom mana yang memberikan sinyal low. Misal tombol angka 1 ditekan, maka saat baris 1 diberikan sinyal low, maka kolom 1 akan memberikan sinyal low, sedang dua tombol lainnya pada baris 1 (tombol angka 2 dan 3) yang tidak ditekan akan terbuka.



Gambar 8-1 Keypad beserta rangkaian internalnya

8.1.1 Rangkaian



Gambar 8-2 Koneksi keypad dengan mikrokontroler

Program 8-1 Program keypad.bas

```
'=====
'Program keypad1.bas
'Program untuk menampilkan tombol keypad yang ditekan
'pada LCD
'=====
Bar1 Alias P2.0
Bar2 Alias P2.1
Bar3 Alias P2.2
Bar4 Alias P2.3
Kol1 Alias P2.4
Kol2 Alias P2.5
Kol3 Alias P2.6
Keyport Alias P2
Dim Keydat As Byte

'=====
'inisialisasi lcd
Config Lcd = 16 * 2
Config Lcdpin = Pin , Db4 = P0.4 , Db5 = P0.5 , Db6 = P0.6 ,
Config Lcdpin = Pin , Db7 = P0.7 , E = P0.2 , Rs = P0.0
P0.1 = 0
Lcdinit
Cls
'=====
=====

Do
Cls
Gosub Keypad
Lcd Keydat
Waitms 500
Loop

End

Keypad:
Keyport = &HFF

Baris1:
Bar1 = 0
Bar2 = 1
Bar3 = 1
Bar4 = 1
If Kol1 = 0 Then
    Keydat = 1
Elseif Kol2 = 0 Then
    Keydat = 2
Elseif Kol3 = 0 Then
    Keydat = 3
Else
    Goto Baris2
End If

Baris2:
```

```

Bar1 = 1
Bar2 = 0
Bar3 = 1
Bar4 = 1
If Koll = 0 Then
  Keydat = 4
  Elseif Kol2 = 0 Then
    Keydat = 5
  Elseif Kol3 = 0 Then
    Keydat = 6
  Else
    Goto Baris3
End If

```

```

Baris3:
Bar1 = 1
Bar2 = 1
Bar3 = 0
Bar4 = 1
If Koll = 0 Then
  Keydat = 7
  Elseif Kol2 = 0 Then
    Keydat = 8
  Elseif Kol3 = 0 Then
    Keydat = 9
  Else
    Goto Baris4
End If

```

```

Baris4:
Bar1 = 1
Bar2 = 1
Bar3 = 1
Bar4 = 0
If Koll = 0 Then
  Keydat = 10
  Elseif Kol2 = 0 Then
    Keydat = 0
  Elseif Kol3 = 0 Then
    Keydat = 11
End If

```

```
Return
```

8.1.2 Prinsip Kerja Program Keypad.bas

Pada dasarnya untuk mengetahui tombol keypad yang ditekan digunakan cara scanning, yang pada program ini dilaksanakan oleh fungsi keydata. Pernyataan Alias pada awal program digunakan untuk mendefinisikan pin yang terhubung dengan baris atau kolom, sehingga jika rangkaian yang menghubungkan mikrokontroler dan keypad berubah, cukup diganti di awal program menggunakan pernyataan Alias.

Pin yang terhubung dengan baris yaitu Bar1, Bar2, Bar3, dan Bar4 merupakan output, yang pada kondisi normal mengeluarkan logika 1 sedangkan Koll1, Kol2, dan Kol3 berfungsi sebagai input. Pernyataan

```

Baris1:
Bar1 = 0           'baris 1 mengeluarkan logika 0
Bar2 = 1

```

44 Belajar Pemrograman Mikrokontroler AT89S51 Menggunakan Bascom-8051

```
Bar3 = 1
Bar4 = 1
If Kol1 = 0 Then                                'cek kolom 1, jk kolom 1 = 0 maka
  Keydat = 1                                    'tombol angka 1 yang ditekan
Elseif Kol2 = 0 Then                             'cek kolom 2, jk kolom 2 = 0 maka
  Keydat = 2                                    'tombol angka 2 yang ditekan
Elseif Kol3 = 0 Then                             'cek kolom 3, jk kolom 3 = 0 maka
  Keydat = 3                                    'tombol angka 3 yang ditekan
Else                                             'jk belum ada data
  Goto Baris2                                    'lanjutkan ke baris 2
End If
```

menunjukkan baris1 mengeluarkan logika 0, yang selanjutnya dicek pada masing-masing kolom apakah ada yang menerima logika 0, karena jika tombol keypad ditekan akan terhubung dengan baris1. Jika masih belum ada kolom yang menerima logika 0, maka scanning dilanjutkan ke baris2, baris3, dan baris4 hingga ditemukan kolom dengan logika 0. Jika telah ditemukan tombol yang ditekan, data tombol akan ditampilkan pada LCD.

Latihan 9

Buatlah program untuk menampilkan 3 konfigurasi gerakan LED yang berbeda, jika tombol 1, 2, atau 3 ditekan. Konfigurasi akan dimulai jika tombol 9 ditekan dan akan berhenti saat tombol 7 ditekan. Tentukan sendiri variasi gerakannya.

9. ADC

Tujuan : Setelah melaksanakan modul ini mahasiswa diharapkan dapat :

- Menjelaskan parameter-parameter pada konversi analog ke digital
- Menyusun rangkaian ADC0804 sekaligus membuat program yang dapat menampilkan data hasil konversi dari analog ke digital
- Membuat pengukur temperatur digital

9.1 Konversi Sinyal Analog Menjadi Digital Menggunakan ADC0804

Selain berupa sinyal digital, sinyal analog juga dapat menjadi input ataupun output dari mikrokontroler. Sinyal input digital haruslah diubah terlebih dahulu menjadi sinyal digital menggunakan ADC (Analog – Digital Converter). Terdapat berbagai cara untuk konversi sinyal analog-digital, yang paling sederhana adalah dengan menggunakan IC ADC. Terdapat berbagai macam jenis IC ADC, yang dapat dibedakan menjadi :

- Input tunggal atau multipleks
ADC dengan input tunggal (misal ADC0804) hanya dapat menerima satu sinyal input analog, sedangkan jenis yang lain adalah jenis multipleks yang dapat menerima beberapa input analog (4,8, atau 16 input), misal ADC0800/09 dan ADC0816/17.
- Resolusi Output
ADC 08XX mempunyai resolusi 8 bit (256 keadaan output). Resolusi dapat dipertinggi dengan menggunakan 10 bit (1024 keadaan output) atau 12 bit (4096 keadaan output)
- Output serial atau paralel
ADC dengan output paralel menyediakan jalur data terpisah untuk masing-masing bit, contohnya adalah ADC0804, ADC0808/09, dan ADC0816/17. Pada ADC dengan output serial hanya terdapat satu jalur data, terkadang ditambah satu jalur clock, sehingga hanya terdapat 1 bit data yang dikirimkan dalam satu waktu, contohnya ADC0831/32/34

Resolusi adalah perubahan terkecil dari tegangan analog input yang dapat dibedakan oleh suatu ADC. Resolusi tergantung pada jumlah keadaan yang mungkin pada output, yang ditentukan oleh jumlah bit, dan rentang tegangan input (atau sering disebut juga sebagai tegangan referensi), yang merupakan selisih dari nilai maksimum tegangan referensi dikurangi nilai minimumnya.

$$N = 2^n$$

N = jumlah keadaan yang mungkin pada output

n = jumlah bit

$$Q = \frac{(V_{\max} - V_{\min})}{N - 1}$$

Q = resolusi, Volt

V_{max} = tegangan referensi maksimum

V_{min} = tegangan referensi minimum

Jika tegangan referensi yang dipergunakan adalah 5 Volt, maka resolusi ADC 0804 adalah

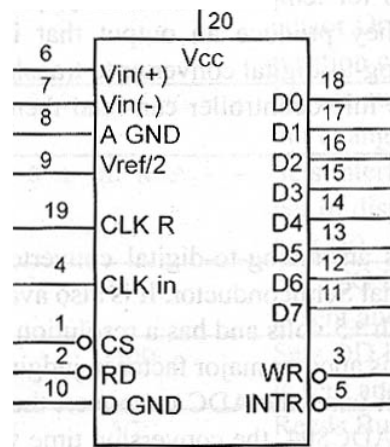
$$Q = \frac{(5V - 0V)}{(2^8) - 1} = \frac{5V}{255} = 0,0196V = 19.6mV$$

Tabel 9-1 menunjukkan hubungan tegangan input dengan output digital pada ADC 8 bit untuk rentang tegangan input 0 – 5 V.

Tabel 9-1 Hubungan antara tegangan input analog dengan output digital

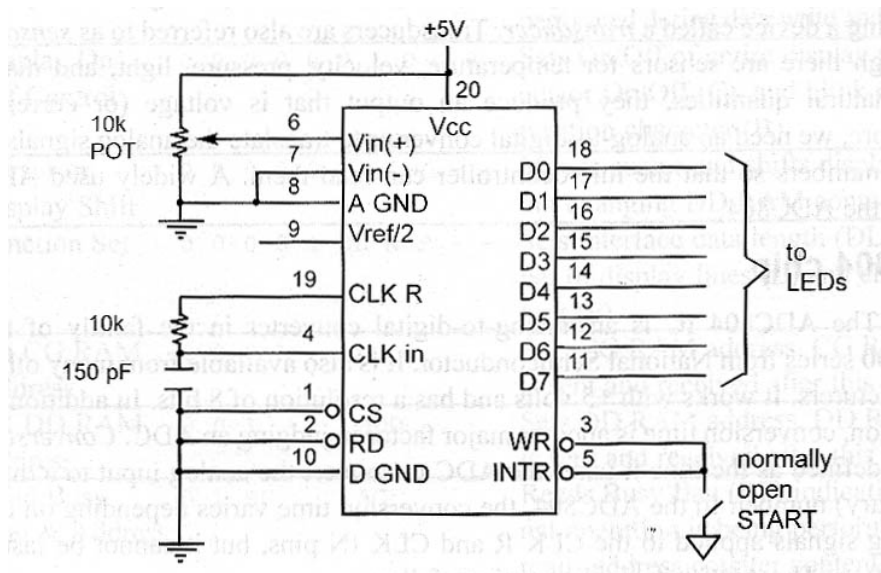
Tegangan input (V)	Output digital (desimal)	Output digital (biner)
0	0	00000000
0,0196	1	00000001
0,0392	2	00000010
0,0588	3	00000011
4,9412	252	11111100
4,9608	253	11111101
4,9804	254	11111110
5	255	11111111

Pada kegiatan ini ADC yang dipergunakan adalah ADC0804, yang merupakan ADC dengan satu input analog dan mempunyai ouput paralel dengan resolusi 8 bit. Skema pin ADC0804 dapat dilihat pada gambar 9-1.



Gambar 9-1 Susunan pin ADC0804

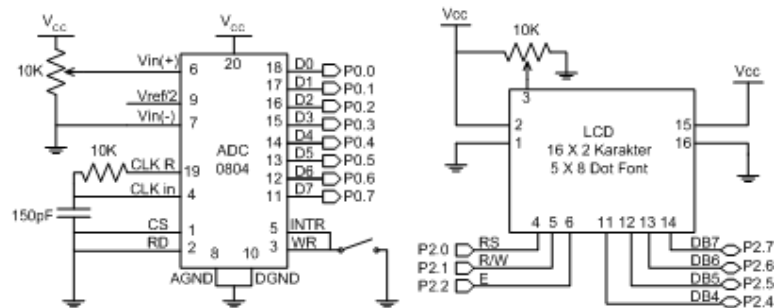
9.1.1 Rangkaian



Gambar 9-2 Rangkaian ADC0804 free running mode.

Rangkaian ADC0804 dapat dibuat bekerja pada free running mode, yaitu bekerja secara terus menerus. ADC0804 memerlukan clock, yang bisa diambil dari mikrokontroler ataupun memanfaatkan clock internal, dengan menambahkan resistor 10 k dan kapasitor 150pF.

Tegangan input yang diukur adalah selisih antara Vin(+) dengan Vin(-). Adapun Vref/2 menunjukkan setengah dari nilai tegangan referensi yang dipergunakan. Pada Gambar 9-2 Vin(-) dihubungkan dengan ground, sedangkan Vref/2 tidak terhubung, sehingga nilai tegangan input adalah Vin(+) dan tegangan referensi nilainya sama dengan Vcc, yaitu 5 V.



Gambar 9-3 Diagram pengkabelan rangkaian ADC0804 free running mode.

Program 9-1 Program adc_1.bas

```

'=====
=====
'Program adc_1.bas
'Program untuk menampilkan hasil konversi ADC pada LCD
'=====
=====

Dim Adc As Byte

Config Lcd = 16 * 2
Config Lcdpin = Pin , Db4 = P2.4 , Db5 = P2.5 , Db6 = P2.6 ,
Config Lcdpin = Pin , Db7 = P2.7 , E = P1.1 , Rs = P1.0
Initlcd

Cls
Cursor Off

Do

Cls
Adc = P0                               'mengambil data hasil
konversi
Lcd Adc                                'menampilkan pada LCD

Waitms 250                             'penundaan supaya
dapat terbaca
Loop

End
'=====

```

48 Belajar Pemrograman Mikrokontroler AT89S51 Menggunakan Bascom-8051

Dengan memutar potensiometer dapat disimulasikan perubahan tegangan analog pada input ADC yang kemudian ditampilkan hasil konversinya pada LCD.

Latihan 10

3. Modifikasi Program 9-1 sehingga LCD dapat menampilkan nilai tegangan pada input ADC (Petunjuk : lihat penjelasan pada pengantar modul ini).
4. Baca pada help Bascom mengenai instruksi FUSING. Buatlah program yang dapat menampilkan nilai tegangan dengan ketelitian 0,01 V.
5. Aturlah potensiometer untuk mendapatkan variasi tegangan analog. Gunakan multimeter untuk mengukur nilai tegangan tersebut. Catat pada lembar aktifitas dan bandingkan dengan nilai yang ditampilkan pada LCD.

9.2 Kegiatan 2 Membuat Termometer Digital

LM35 adalah sensor temperatur terkalibrasi untuk $^{\circ}\text{C}$ yang menghasilkan output sebesar $0,01 \text{ V}/^{\circ}\text{C}$ dengan rentang pengukuran $-55 - 150 ^{\circ}\text{C}$. Gambar 9-4 menunjukkan susunan pin dari LM35.



Gambar 9-4 Susunan pin LM35

Latihan 11

Gantikan potensiometer dengan sensor temperatur LM35. Tampilkan temperatur pada LCD dengan format penulisan “Temperatur” pada baris pertama dan “XX.X 0C” pada baris kedua.

10. Komunikasi Serial Sinkron

Tujuan : Setelah melaksanakan modul ini mahasiswa diharapkan dapat :

- Membedakan komunikasi serial dan paralel
- Memanfaatkan register geser untuk menambah jumlah input dan output

Data dapat ditransmisikan dengan dua cara, yaitu paralel dan serial. Di antara keduanya transmisi data paralel mempunyai kecepatan tertinggi, namun juga mempunyai kelemahan, yaitu memerlukan banyak jalur I/O dan hanya dapat digunakan untuk jarak yang relatif dekat. Transmisi data serial hanya memerlukan jalur I/O yang sedikit, biasanya berjumlah 1, 2, atau 3 jalur I/O.

Terdapat dua metode yang dapat digunakan pada transmisi data serial, yaitu sinkron dan asinkron. Pada metode sinkron, selain jalur data terdapat satu jalur tambahan yang dipergunakan sebagai clock, sedangkan pada metode asinkron clock dibangkitkan secara terpisah pada masing-masing pengirim dan penerima.

Kali ini kita akan memanfaatkan komunikasi serial untuk menambah jalur I/O pada mikrokontroler dengan menggunakan register geser (shift register). Pada register geser output data masuk secara seri (bit demi bit) untuk diubah menjadi output paralel (SIPO = Serial In Paralel Out), sedangkan pada register geser input data dimasukkan secara paralel untuk dikeluarkan secara seri (PISO = Paralel In Serial Out). Register geser output yang dipergunakan adalah 74HC595 sedangkan register geser input menggunakan 74HC165.

10.1 Kegiatan 1 : Penggunaan Register Geser Untuk Menambah Jumlah Output

Pada kegiatan ini dipergunakan 3 buah pin output untuk mengendalikan 8 output berupa LED.

Program

```
'=====
'Program reg_out.bas
'Program untuk mengendalikan 8 LED menggunakan 3 pin I/O
'=====
Pclock Alias P0.0           'P0.0 sebagai pin clock
Pdat Alias P0.1             'P0.1 sebagai pin data
Platch Alias P0.2          'P0.2 sebagai pin latch

Dim A As Byte

Reset Platch                'latch off

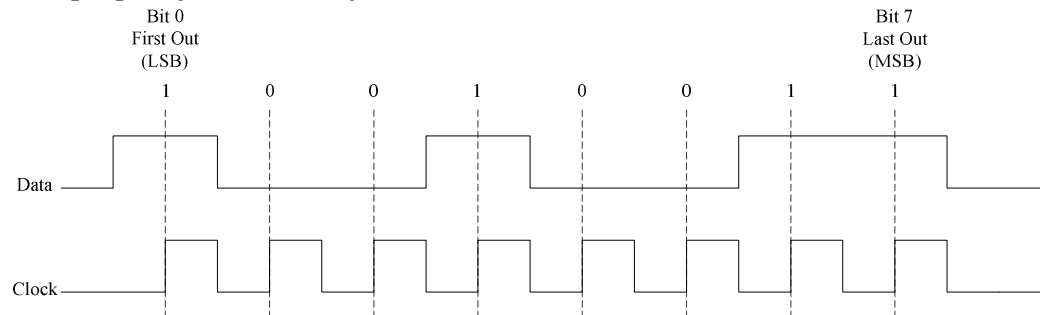
Do
A = &H0F                    'data yang akan dimasukkan &H0F
Gosub Out_595              'panggil subrutin
Waitms 500                 'tunggu 0,5 detik
A = &HF0                    'data yang akan dimasukan &HF0
Gosub Out_595              'panggil subrutin
Waitms 500                 'tunggu 0,5 detik
Loop

End
```

50 Belajar Pemrograman Mikrokontroler AT89S51 Menggunakan Bascom-8051

```
Out_595:                                'subrutin untuk mengirim data ke
74HC595
Shiftout Pdat , Pclock , A , 0         'kirim data ke 74HC595
Set Platch                               'output          di-latch          untuk
ditampilkan
Waitms 1
Reset Platch
Return
```

Pada komunikasi serial sinkron diperlukan minimal 2 jalur I/O, yaitu data dan clock. Pada rangkaian ini clock diberikan oleh mikrokontroler. Gambar 10-1 menunjukkan sinyal yang terdapat pada jalur data dan jalur clock.



Gambar 10-1 Ilustrasi komunikasi asinkron

Pengiriman data secara serial ke 74HC595 dilakukan dengan menggunakan pernyataan shiftout. Pernyataan ini mempunyai sintaks

```
SHIFTOUT pin , pclock , var , option
```

pin adalah pin yang dipergunakan sebagai output data

pclock adalah pin yang dipergunakan sebagai jalur clock

var adalah data yang akan dikirimkan

option pilihan cara pengiriman yang terdiri dari

0 - MSB dikirim dahulu saat logika clock low

1 - MSB dikirim dahulu saat logika clock high

2 - LSB dikirim dahulu saat logika clock low

3 - LSB dikirim dahulu saat logika clock high

Pernyataan

```
Shiftout Pdat , Pclock , A , 0
```

menunjukkan bahwa Pdat (P0.1) digunakan sebagai output data, Pclock (P0.0) sebagai jalur clock, dengan data yang dikirim adalah variabel A dan yang dikirim adalah MSB terlebih dahulu saat logika clock low.

Pada 74HC595 setelah proses pengiriman serial selesai data haruslah di-latch terlebih dahulu sebelum data dikeluarkan secara paralel. Hal ini dilakukan dengan cara memberikan logika low-high-low melalui Platch (P0.2) seperti terdapat pada pernyataan

```
Set Platch
```

```
Waitms 1
```

```
Reset Platch
```

74HC595 dapat juga digunakan dalam jumlah lebih dari satu. Gb menunjukkan contoh rangkaian untuk menghubungkan 2 buah 74HC595 dengan menggunakan 3 pin I/O. Tidak ada batasan jumlah 74HC595 yang dapat dihubungkan ke mikrokontroler, namun yang perlu diperhatikan adalah semakin banyak jumlah 74HC595 yang dihubungkan, akan memperlambat kecepatan program, karena semakin banyak data yang dikirim secara serial akan memakan waktu lebih lama.

11. Komunikasi Serial Asinkron Dengan PC

Tujuan : Setelah melaksanakan modul ini mahasiswa diharapkan dapat :

- Melakukan komunikasi antara mikrokontroler dengan PC

Transmisi data serial secara asinkron hanya memerlukan 1 jalur I/O. Pada komunikasi serial terdapat kesulitan dalam menentukan awal dan akhir data, kecuali pengirim dan penerima menerapkan perangkat aturan yang sama yang disebut protokol. Protokol mengatur bagaimana data dibentuk, berapa banyak bit untuk membentuk suatu karakter, serta kapan data mulai atau berhenti.

Pada komunikasi asinkron, data diletakkan di antara start bit dan stop bit. Gambar menunjukkan saat tidak ada transfer data sinyal berada pada kondisi high, atau disebut mark. Transmisi dimulai dengan start bit, yang selanjutnya diikuti dengan data yang dikirim berturut-turut mulai LSB hingga MSB, dan kemudian diakhiri dengan stop bit.

Pada komunikasi asinkron peranti dapat diprogram untuk data dengan lebar data 7 atau 8 bit, sedangkan stop bit dapat berjumlah 1 atau 2. Saat ini pengiriman data yang umum dilakukan berupa data ASCII 8 bit dan diakhiri dengan 1 buah stop bit. Sehingga untuk satu karakter total bit yang dikirim adalah 10 bit, 8 bit untuk karakter serta masing-masing 1 bit untuk start bit dan stop bit.

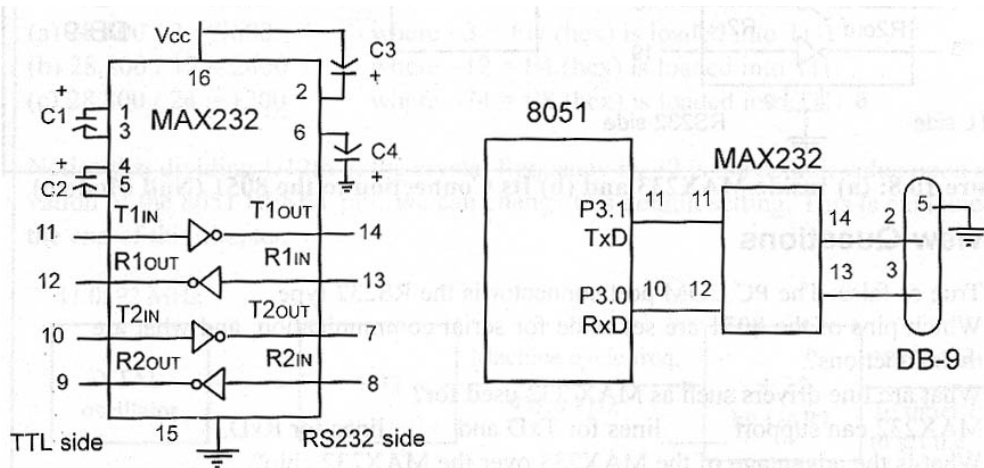
Pada beberapa sistem, untuk memeriksa validitas data yang dikirim, digunakan bit tambahan yang disebut parity bit. Parity bit tersebut bisa berupa parity bit ganjil ataupun genap. Untuk parity bit genap jumlah keseluruhan bit termasuk parity bit adalah genap, sedangkan untuk parity bit ganjil jumlah keseluruhan bit adalah ganjil. Contohnya untuk karakter "A" dengan kode ASCII 01000001 akan mempunyai parity bit genap 0 atau 1 untuk parity bit ganjil.

Terdapat bermacam-macam standar komunikasi asinkron, salah satu yang paling populer adalah RS 232. Pada RS232 logika 1 direpresentasikan oleh tegangan antara - 3V hingga - 25V, sedangkan logika 0 direpresentasikan oleh tegangan antara + 3V hingga +25V, dengan level tegangan paling umum dipakai adalah - 12V dan + 12 V. Karena level tegangan yang dipergunakan RS232 tidak kompatibel dengan level tegangan TTL yang dipakai mikrokontroler (5 V untuk logika 1 dan 0 V untuk logika 0), maka diperlukan konverter tegangan, yaitu menggunakan IC MAX 232.

Untuk dapat berkomunikasi pengirim dan penerima haruslah mengirim dan menerima data dengan kecepatan transfer data yang sama. Kecepatan transfer ini disebut baud rate yang menyatakan banyaknya perubahan sinyal per detik. Nilai baud rate yang sering dipakai adalah 1200, 2400, 4800, dan 9600.

11.1.1 Rangkaian

Sebagai penghubung pada komunikasi serial dipergunakan kabel serial menggunakan soket DB9 yang mempunyai 9 pin. Untuk koneksi paling sederhana (null mode) cukup digunakan 3 pin, yaitu TxD, RxD, dan GND.

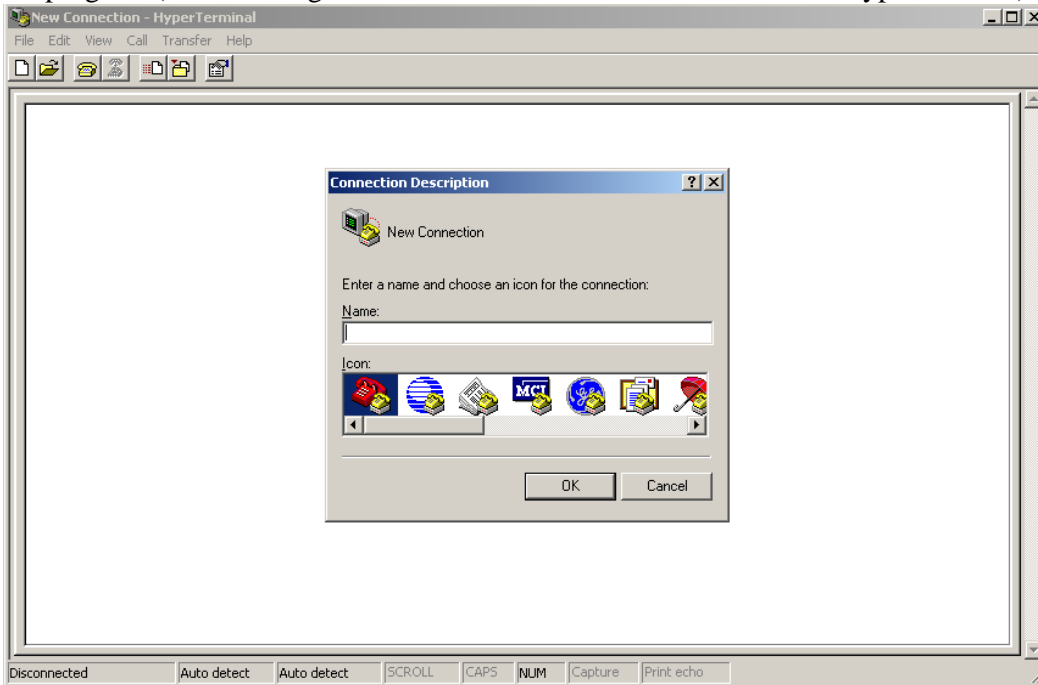


Gambar 11-1 Rangkaian konverter TTL – RS232 menggunakan MAX232

Pada MAX232 terdapat 2 pin yang digunakan untuk transfer data, yaitu TxD untuk mengirimkan data dan RxD untuk menerima data. Pada mikrokontroler pin yang dipergunakan adalah pin P3.1 (sebagai TxD) dan pin 3.0 (sebagai RxD). Hubungan antara MAX232 dan mikrokontroler terdapat pada gambar. Supaya dapat bekerja baik rangkaian ini memerlukan 4 buah kapasitor dengan nilai antara 1 hingga 22 μ F.

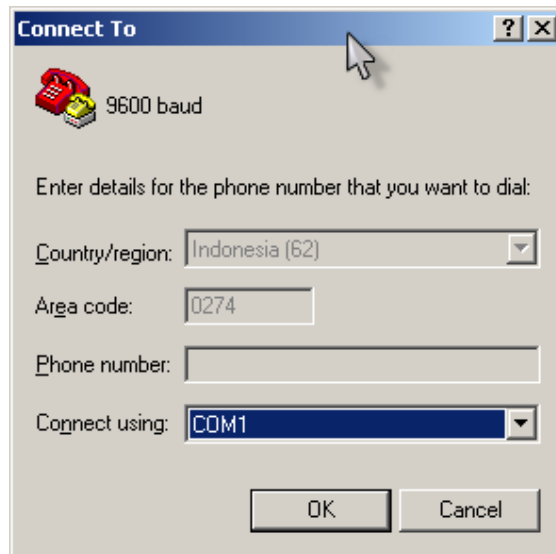
Untuk menampilkan data karakter yang dikirim mikrokontroler di PC digunakan program Hyperterminal, yaitu salah satu program yang terdapat pada MS Windows. Untuk menjalankan program Hyperterminal dilakukan dengan urutan berikut

Buka program (Start \rightarrow Program \rightarrow Accesories \rightarrow Communications \rightarrow Hyperterminal).



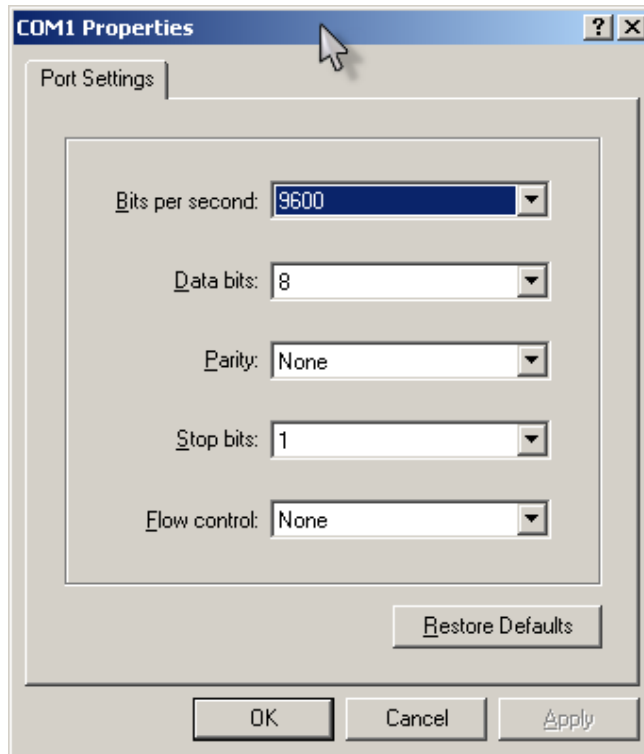
Gambar 11-2 Tampilan awal program Hyperterminal

Pada jendela Connection Description isikan nama untuk koneksinya, misalnya mikrokontroler 9600 baud. Klik OK.



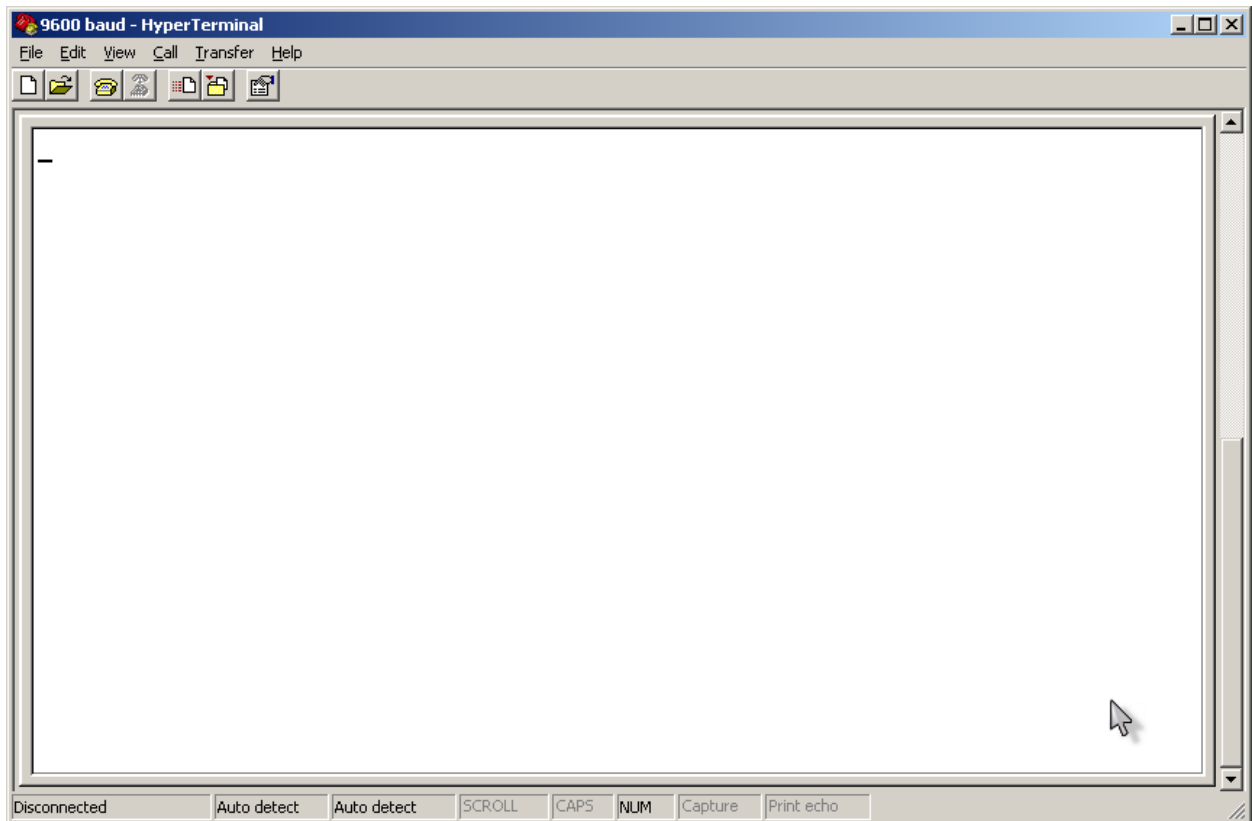
Gambar 11-3 Jendela Connection Description

Pada jendela connect to isikan connect using dengan port serial yang akan dipergunakan (COM 1).



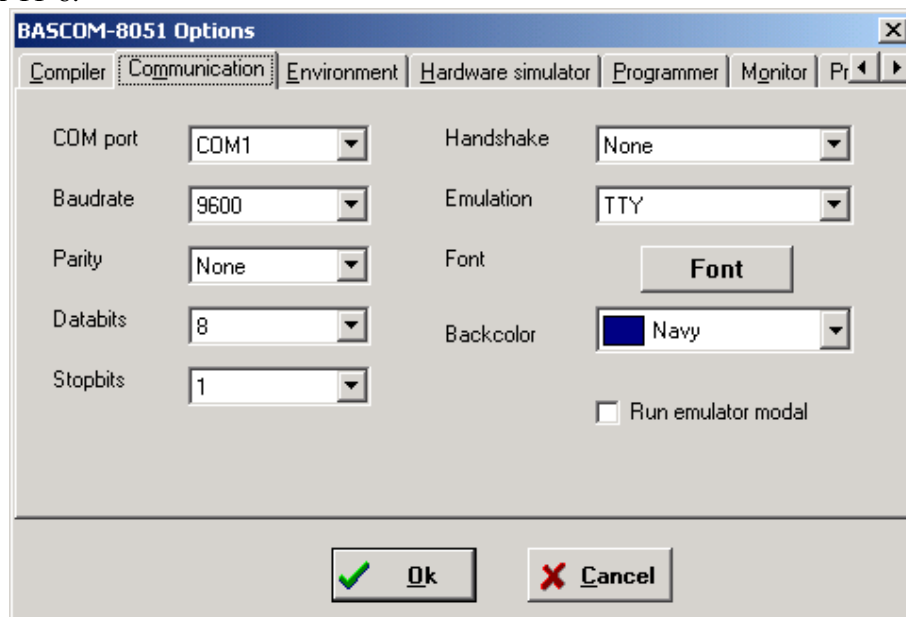
Gambar 11-4 Jendela COM1 properties

Pada jendela COM1 Properties masukkan parameter komunikasi yang diinginkan yaitu 9600 bits per detik, 8 bit data, tanpa bit parity, 1 bit stop, dan tidak ada flow control. Klik OK maka Hyperterminal siap untuk dipergunakan.



Gambar 11-5 Tampilan Hyperterminal yang siap untuk digunakan

Syarat utama terjadinya komunikasi asinkron adalah parameter komunikasi yang digunakan haruslah sama, untuk itu parameter pada mikrokontroler juga harus diatur juga. Pada Bascom klik Option → Communication hingga tampil jendela seperti terlihat pada Gambar 11-6.



Gambar 11-6 Jendela Options – Communication pada Bascom

Program 11-1 Pprogram PC1.bas

```
Print "selamat belajar komunikasi menggunakan port serial"
End
```


Program PC1.bas sangatlah sederhana, yaitu mikrokontroler mengirimkan data berupa kalimat yang ditampilkan pada hyperterminal.

Program 11-2 Program PC2.bas

```
Dim Pilihan As Byte , X As Byte
Do
Print "masukkan pilihan anda"
Print "1"
Print "2"
Print "3"
Input Pilihan

Select Case Pilihan
    Case 1 : Gosub Satu
    Case 2 : Gosub Dua
    Case 3 : Gosub Tiga
    Case Else : Print "anda belum memilih"
End Select

Print "terima kasih atas partisipasi anda"

Loop

Satu:
Print "pilihan anda no satu"
Return

Dua:
Print "pilihan anda no dua"
Return

Tiga:
Print "pilihan anda no tiga"
Return
```

Program 11-2 menunjukkan contoh komunikasi antara mikrokontroler dengan PC. Di sini PC dimanfaatkan sebagai media input dan output yang langsung terhubung dengan pengguna.

Latihan 12

Buatlah program untuk menampilkan hasil pengukuran temperatur menggunakan Hyperterminal yang diawali dengan pertanyaan kepada pengguna, skala temperatur apa yang akan digunakan, apakah dalam Celsius, Kelvin, Fahrenheit, atau Rankine.

12. Aktuator

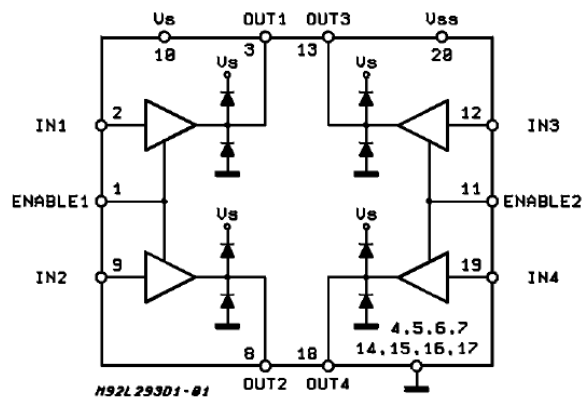
Tujuan : Setelah melaksanakan modul ini mahasiswa diharapkan dapat :

- Membuat program yang dapat mengendalikan gerak motor DC dan motor servo

12.1 Motor DC

Motor adalah peranti yang mengkonversikan energi elektrik menjadi gerak mekanik, berupa gerak rotasi. Berdasarkan tegangan catu dayanya motor dapat dibedakan menjadi motor DC dan motor AC. Motor DC digunakan pada banyak aplikasi sistem mekatronika karena relatif mudah dalam pengendaliannya dibandingkan motor AC. Kecepatan motor DC dapat diatur dan semua motor DC putarannya dapat dibalik.

Untuk mengendalikan motor DC diperlukan suatu rangkaian yang di dalamnya terdapat dari transistor ataupun relay. Di pasaran juga terdapat rangkaian terpadu pengendali motor, seperti misalnya L293D. IC ini mempunyai tiga buah input, yaitu IN1, IN2, dan Enable. Pada IC ini terdapat dua jembatan-H yang dapat digunakan untuk dua buah motor. Sebuah rangkaian H-bridge terdiri dari 2 buah pin input (IN1 dan IN2 atau IN3 dan IN4), 1 pin enable (ENABLE1 atau ENABLE2) dan 2 buah pin output (OUT1 dan OUT2 atau OUT3 dan OUT4).



Gambar 12-1 L293D

Output L293 mempunyai tiga keadaan, yaitu logika tinggi dan rendah serta impedansi tinggi. Pada saat impedansi tinggi (Enable = Low) output seakan-akan berfungsi sebagai saklar terbuka. Jika Enable diberi logika tinggi maka output akan tergantung pada IN1 dan IN2. Jika OUT1 dan OUT2 masing-masing dihubungkan dengan terminal pada motor IC ini dapat digunakan untuk mengendalikan motor. Tabel kebenaran L293D jika digunakan untuk mengendalikan motor dapat dilihat pada Tabel 12-1. L293D dapat beroperasi pada tegangan kerja 4,5 V – 36 V dengan arus 600 mA pada kondisi normal dan 1,2 A saat beban puncak sesaat dengan kecepatan maksimal pensaklaran adalah 5 kHz.

Tabel 12-1 Tabel kebenaran L293 untuk pengendalian motor

	Input	Fungsi
Enable = H	Input 1 = H Input 2 = L	Maju
	Input 1 = L Input 2 = H	Mundur
	Input 1 = Input 2	Pengereman

Enable = L	Input 1 = X Input 2 = X	Berhenti
------------	----------------------------	----------

Program 12-1 Program DC1.bas

```
En2 Alias P0.4
In3 Alias P0.3
In4 Alias P0.2
```

```
P0 = 0
```

```
Do
Gosub Berhenti
Wait 2
Gosub Maju
Wait 2
Gosub Berhenti
Wait 2
Gosub Mundur
Wait 2
Loop
End
```

```
Maju:
En2 = 1
In3 = 1
In4 = 0
Return
```

```
Mundur:
En2 = 1
In3 = 0
In4 = 1
Return
```

```
Berhenti:
En2 = 0
In3 = 0
In4 = 0
Return
```

Program ini akan memerintahkan motor untuk berputar maju 2 detik dan berputar mundur 2 detik, masing-masing diselingi berhenti selama 2 detik. Perhatikan masing-masing subrutin Manju, Mundur, dan Berhenti yang sesuai dengan tabel 12-1.

Latihan 13

Buatlah program untuk mengatur arah putaran motor DC menggunakan keypad. Fungsi tombol 5 untuk menghentikan motor, tombol 8 untuk berputar maju, dan tombol 2 untuk berputar mundur.

12.2 Motor Servo

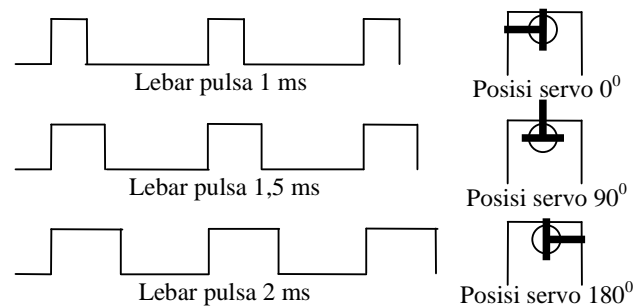
Motor servo adalah salah satu jenis motor DC yang beroperasi secara closed loop. Poros motor dihubungkan dengan dengan rangkaian kendali, sehingga jika putaran poros belum sampai pada posisi yang diperintahkan maka rangkaian kendali akan terus mengkoreksi

posisi hingga mencapai posisi yang diperintahkan. Motor servo banyak digunakan pada peranti R/C (remote control) seperti mobil, pesawat, helikopter, dan kapal, serta sebagai aktuator robot maupun penggerak pada kamera.



Gambar 12-2 Motor servo

Motor servo terdiri dari dua jenis yaitu motor servo standar yang hanya dapat bergerak pada rentang sudut tertentu, biasanya 180° atau 270° , dan motor servo kontinu yang dapat berputar secara kontinu. Pada motor servo standar yang dapat dikendalikan adalah posisi poros sedangkan pada motor servo kontinu yang dapat dikendalikan adalah kecepatan. Cara pengendaliannya adalah sama yaitu dengan mengatur lebar pulsa yang diberikan. Lebar pulsa yang diperlukan antara 1 ms hingga 2 ms. Gambar 12-3 menunjukkan hubungan antara lebar pulsa yang diberikan dengan posisi poros. Pulsa haruslah selalu diulang setiap 20 hingga 30 ms atau frekuensi kurang lebih 50 Hz.



Gambar 12-3 Hubungan antara lebar pulsa dengan posisi poros motor servo

Program 12-2 Program Servo1.bas

```
Config Servos = 1 , Servo1 = P1.1 , Reload = 100
Servo1 = 15
End
```

Program ini menunjukkan dasar pengendalian motor servo menggunakan fungsi yang telah ada pada Bascom. Config Servos menunjukkan jumlah motor servo yang terhubung dengan mikrokontroler. Servo1 = P1.1 menunjukkan pin yang digunakan. Reload = 100 menunjukkan waktu reload dalam mikrodetik. Servo1 = 15 menunjukkan pulsa yang dikirim selama 15×100 mikrodetik, atau 1,5 milidetik.

Daftar Pustaka

- Agung Nugroho Adi, Handout Kuliah Mekatronika, Teknik Mesin UII, 2007.
- Alciatore, D. G. & Hestand, M. B., Introduction to Mechatronics and Measurement Systems, McGraw-Hill, 2003.
- Anonim, Bascom-8051 Help Reference, MCS Electronics, 2006.
- Didin Wahyudin, Belajar Mudah Mikrokontroler AT89S52 dengan Bahasa BASIC Menggunakan BASCOM-8051, Penerbit Andi Yogyakarta, 2006.
- Mazidi, M. A. & Mazidi, J. G., The 8051 Microcontroller And Embedded Systems, Prentice Hall, 2000.
- McComb, G & Predko, M., Robot Builder's Bonanza, McGraw-Hill, 2006.
- Muljowidodo K. & Indra Djodikusumo, Mekatronika, HEDS ITB, 1996.
- Stiffler, A. K., Design with Microprocessor for Mechanical Engineers, McGraw-Hill, 1992.
- Tokheim, R. L., Digital Electronics, Principles and Applications, McGraw-Hill, 2003.
- Wahyudi, Handout Kuliah Kontrol Berbasis Komputer, Teknik Mesin UII, 2004.
- Simpson, Colin D., Industrial Electronics, Prentice-Hall, 1996.