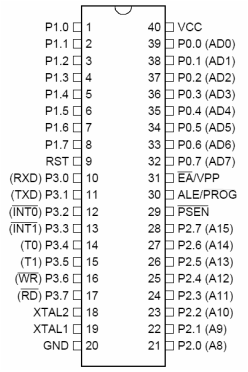
BAB II
LANDASAN TEORI

2.1 Mikrokontroller AT89S51

Atmel 89S51 adalah mikrokontroller yang dikeluarkan oleh Atmel pada tahun 1994, sebuah perusahaan yang bergerak dalam produksi semikonduktor, dengan arsitektur MCS51 dengan 4Kb Flash PEROM (*Programmable and Erasable Read Only Memori*) yaitu ROM yang dapat ditulis dan dihapus menggunakan perangkat programmer, juga memiliki 128 x 8 bit internal RAM, 32 *Programmable I/O lines* dan mempunyai 40 kaki, 32 kaki diantaranya adalah kaki untuk keperluan port parallel. Satu port paralel terdiri dari 8 kaki, dengan demikian 32 kaki tersebut membentuk 4 buah port paralel, yang masing-masing dikenal sebagai Port-0, Port-1, Port-2, Port-3. Nomor dari masing-masing jalur/kaki dari port paralel mulai dari 0 sampai 7. Jalur atau kaki pertama Port-0 disebut sebagai P0.0 dan jalur terakhir untuk Port-3 adalah P3.7.





(a) (b)

Gambar 2.1 (a) Mikrokontroller AT89S51 dan (b) struktur kaki Mikrokontroller AT89S51

Adapun nama dan fungsi dari kaki (pin) pada mikrokontroller AT89S51 adalah sebagai berikut :

1. *RST (Reset)*

   Masukan reset kondisi 1 selama 2 siklus mesin selama osilator bekerja akan me-*reset* mikrokontroller yang bersangkutan ke alamat awal.

2. *ALE/PROG (Address Latch Enable)*

   Keluaran ALE atau *Address Latch Enable* menghasilkan pulsa-pulsa untuk mengunci bit rendah (*low byte*). Selama mengakses memori eksternal, kaki ini juga berfungsi sebagai masukan pulsa program (*the program pulse input*) atau PROG selama pemograman *flash*. Pada operasi normal, ALE akan berpulsa dengan laju 1/6 dari frekuensi kristal dan dapat digunakan sebagai pewaktuan (*timing*) atau pendetakan (*clocking*) rangkaian eksternal, ada satu pulsa yang dilompati selama pengaksesan memori data eksternal. Jika dikehendaki operasi, ALE bisa dimatikan dengan cara mengatur bit 0 dari SFR lokasi 8EH, jika isinya 1, ALE hanya akan aktif selama dijumpai intruksi MOVX atau MOVC. Selain itu kaki ini akan secara lemah di-*pulled-high*, mematikan bit ALE tidak akan ada efeknya jika mikrokontroller mengeksekusi program secara external bisa juga untuk me-*multiplexing* antara *address* dan address data *bus*.

3. *PSEN*

   Program Store Enable merupakan sinyal baca untuk memori program eksternal, saat mikrokontroller keluarga 51 menjalankan program dari memori eksternal, PSEN akan diaktifkan dua kali persiklus mesin, kecuali dua aktifasi PSEN dilompati (diabaikan) saat mengakses memori data eksternal.

4. *EA/Vpp*

   Eksternal acces enable harus selalu dihubungkan ke ground, jika mikrokontroller akan mengeksekusi program dari memori ekternal lokasi 0000H hingga FFFFH, selain itu EA harus dihubungkan ke Vcc agar mikrokontroller mengakses program secara internal. Kaki ini juga berfungsi menerima tegangan 12 Volt(Vpp) selama pemograman *flash*.

5. *Port*

   Berfungsi sebagai I/O atau *address data bus* (*external memori*)

6. *Vcc*

   Berfungsi sebagai suplai tegangan

7. *GND*

   Ground atau pembumian.

2.1.1 Spesifikasi Port pada AT89S51

Port pada AT89S51 merupakan port keluaran atau masukan (I/O), sebagai port keluaran masing-masing kaki dapat menyerap arus (*sink*) delapan masukan TTL (sekitar 3.8 mA). Pada saat '1' dituliskan ke kaki-kaki port ini, kaki-kaki port dapat digunakan sebagai masukan berimpedansi tinggi. Semua port juga dapat dikonfigurasikan sebagai bus alamat atau data bagian rendah (*low byte*) selama pengaksesan memori data dan program eksternal.

2.1.2 Struktur Memori

AT89S51 mempunyai struktur memori yang terdiri dari :

1. RAM Internal

   RAM Internal memiliki memori sebesar 128 byte yang biasanya digunakan untuk menyimpan variable atau data yang bersifat sementara, dialamati oleh *RAM Address Register* (Register Alamat RAM).

   RAM Internal terdiri atas:

   a. Register Banks

   AT89S51 memiliki delapan buah register yang terdiri dari R0 sampai R7 yang terletak pada alamat 00h hingga 07h pada setiap kali reset.

   b. Bit Addressable RAM

   RAM dengan alamat 20H hingga 2FH dapat diakses secara pengalamatan bit (*bit addressable*) sehingga hanya sebuah intruksi saja setiap bit dalam area ini dapat di-*set, clear, AND*, dan *OR*.

   c. RAM Keperluan Umum (General Purpose RAM)

   RAM keperluan umum dimulai dari alamat 30H hingga 7FH dan dapat diakses dengan pengalamatan langsung maupun tak langsung. Pengalamatan langsung dilakukan ketika salah satu operand merupakan bilangan yang dialamati. Sedangkan pengalamatan tak langsung pada lokasi dari RAM Internal ini adalah akses data dari mamori ketika alamat memori tersebut tersimpan dalam suatu register R0 atau R1 yang dapat digunakan sebagai pointer dari lokasi memori pada RAM Internal.

2. *Flash* PEROM

Memori yang digunakan untuk menyimpan intruksi-intruksi MCS-51 di alamati oleh *Program Address Register* (Register Alamat Program). AT89C51 memiliki 4 Kb *Flash* PEROM yang menggunakan *Atmel's High-Density Non Volatile Technology.*

Program yang ada pada *Flash* PEROM akan dijalankan jika pada saat sistem di-*reset*, pin EA/Vp berlogika satu maka mikrokontroller aktif berdasarkan program yang ada pada *Flash* PEROM nya. Namun jika pin EA/Vp berlogika nol, maka mikrokontroller aktif berdasarkan program yang ada pada memori eksternal.

2.2 Bahasa Assembly

Assembler adalah program komputer yang men-translitrasi program dari bahasa *assembly* ke bahasa mesin. Sedangkan bahasa *assembly* adalah ekuivalensi bahasa mesin dalam bentuk alpanumerik. *Mnemonics* alpanumerik digunakan sebagai alat bantu bagi programmer untuk memprogram mesin komputer dari pada menggunakan serangkaian 0 dan 1 (bahasa mesin) yang panjang dan rumit.



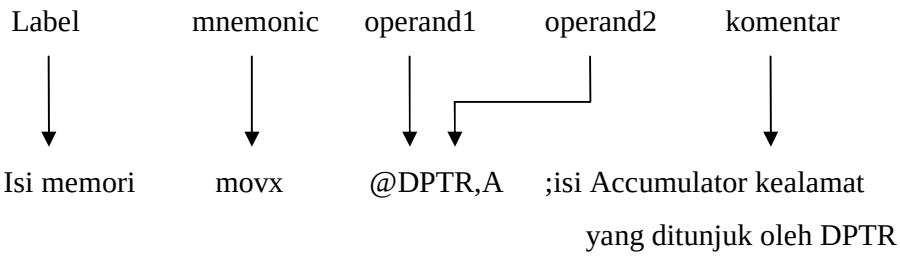
Gambar 2.2 Proses *Assembly*

2.2.1 Kontruksi Program Assembly

Program sumber *assembly* terdiri dari kumpulan baris-baris perintah dan biasanya disimpan dengan extension.ASM dengan satu baris untuk satu perintah, setiap baris perintah tersebut bias terdiri atas beberapa bagian (*field*), yakni bagian label, bagian mnemonic, bagian operand yang bias lebih dari satu dan terakhir bagian komentar. Program sumber (*source code*) dibuat dengan program editor biasa yaitu *Note Pad* pada *Windows*, selanjutnya program sumber diterjemahkan ke bahasa mesin dengan menggunakan program *Assembler*. Hasil kerja program

Assembler adalah *'program objek'* dan juga *'assembly listing'*, dengan ketentuan sebagai berikut :

1. Masing-masing bagian dipisahkan dengan spasi atau TAB, khusus untuk operand yang lebih dari satu masing-masing operand dipisahkan dengan koma.
2. Bagian-bagian tersebut tidak harus semuanya ada dalam sebuah baris, jika ada satu bagian yang tidak ada, maka spasi atau TAB sebagai pemisah bagian tetap harus ditulis.
3. Bagian label ditulis mulai huruf pertama dari baris, jika baris bersangkutan tidak mengandung label, maka label tersebut digantikan dengan spasi atau TAB, yakni sebagai tanda pemisah antara bagian label dan bagian *mnemonic*.



Gambar 2.3 Bentuk Program Sumber Assembler

2.2.2 Assembler Directive

Assembler Directive digunakan untuk mengatur kerja dari program *Assembler*. *Mnemonic* untuk *Assembler Directive* tergantung pada program *Assembler* yang dipakai. Ada beberapa *Assembler Directive* yang umum, yang sama untuk banyak macam program *Assembler* diantaranya adalah :

1. ORG (*Origin*)

   ORG digunakan untuk menyatakan lokasi memori tempat intruksi atau perintah yang ada dibawahnya disimpan.
2. EQU(*Equate*)

   EQU digunakan untuk mendefinisikan sebuah symbol atau lambang *Assembler* secara bebas.
3. DB(*Define Byte*)

DB digunakan untuk memberikan nilai tertentu pada memori program.

4. DW(*Define Word*)

   DW dipakai untuk memberikan nilai 2 byte ke *memory-program* pada baris bersangkutan. *Assembler Directive* ini biasa dipakai untuk membentuk suatu label yang isinya adalah nomor-nomor *memory program.*

5. DS(*Define Storage*)

   Assembler Directive ini dipakai untuk membentuk variable. Sebagai variable tentu saja memori yang dipakai adalah *memory data* (RAM) bukan *memory program* (ROM).

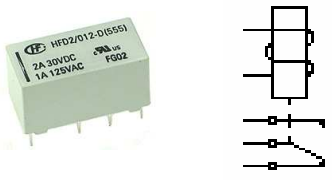
2.3 Komponen Instrumentasi

2.3.1 Relay

Relay adalah sebuah saklar elektromagnetik yang prinsip kerjanya menggunakan azas kumparan listrik.

Prinsip kerjanya adalah sebagai berikut :

Sebuah kumparan yang berintikan sebuah lempengan besi lunak yang apabila dialiri aliran listrik, maka lempengan besi lunak tersebut akan menjadi magnit. Magnit tersebut menarik/menolak pegas kontak sebuah alat penghubung dan akibatnya akan terjadi kontak dan lepas kontak dari alat penghubung tersebut. Prinsip seperti ini dapat dimanfaatkan sebagai dasar pembuatan saklar otomatis yang banyak dipergunakan dalam bidang elektronika.



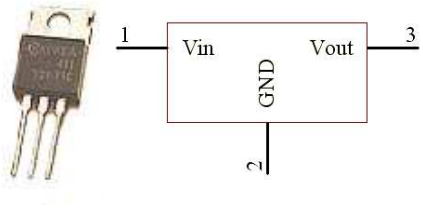
Gambar 2.4 Kontak Relay dan Simbol Relay

2.3.2 Regulator IC LM 78xx dan 79xx

IC dengan tiga kaki yang digunakan sebagai komponen pendukung dari Vcc untuk menghasilkan tegangan 5V, 9V, dan -5V. Tegangan pada transformator tidak selalu tetap atau selalu berubah-ubah, agar transformator stabil maka ditambahkan IC regulator untuk menstabilkan tegangan.

Karakteristik IC regulator :

1. Ada tipe positif untuk meregulasi tegangan positif.
2. Ada tipe negative untuk meregulasi tegangan negative.
3. Penerapan IC mengharuskan Vi>Vo.
4. Dalam praktek, minimum Vi~Vo + 3Volt.
5. Agar dapat menerapkan Vi>Vo perlu menggunakan komponen-komponen lepasan.
6. Transistor dalam konfigurasi tunggal emitor, dimana Vi > Vo hanya kalau transistornya jenuh.



Gambar 2.5 IC Regulator dan Simbol IC Regulator

2.3.3 Transistor Sebagai Saklar

Transistor adalah komponen aktif yang dapat digunakan sebagai penguat sinyal (dalam dunia analog) atau sebagai saklar (dalam dunia digital). Tergantung dari bahan dan media pembuatannya, transistor dapat dibedakan menjadi nMOS (*negative Metal Oxide Semiconductor*), CMOS (*Complementary MOS*), dan TTL (*Transistor-Transistor Logic*). Seperti juga hal nya dioda, transistor juga dibuat dari bahan *Germanium, Silicon,* dan *Indium*. Transistor sendiri sebenarnya adalah hasil pengembangan dari dua buah dioda jenis PN dan NP yang dipertemukan sehingga akan membentuk satu elektroda yang berfungsi sebagai pengontrol pertemuan antara bahan PN dan NP tersebut.

Transistor dapat difungsikan sebagai saklar elektronika dengan memanfaatkan dua keadaan transistor yaitu keadaan saturasi (sebagai saklar tertutup) dan keadaan cut off (sebagai saklar terbuka).

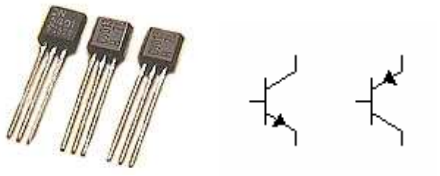
Pada saat saturasi maka arus kolektor adalah:

$$I_{C(sat)} = \frac{VCC}{RC}$$

Pada saat cut off tegangan kolektor emiter sama dengan tegangan sumber kolektor dan arus basis mendekati nol.

$$V_{CE(cut)} = Vcc$$

Untuk komponen transistor dapat dilihat pada gambar dibawah ini

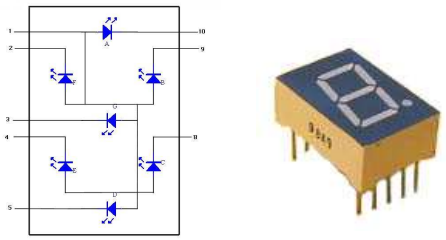


Gambar 2.6 Transistor dan Simbol Transistor

2.3.4 Display 7-Segmen

Display 7-Segmen adalah suatu piranti yang digunakan untuk menampilkan hasil perubahan bilangan biner ke dalam bentuk decimal atau decoder agar langsung dapat dilihat dan dimengerti. Salah satu cara untuk menampilkan angka atau huruf adalah dengan metode matrik.

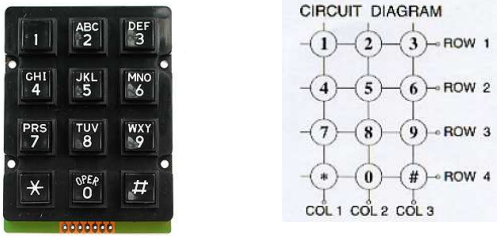
Dengan metode matrik, beberapa garis atau segmen tersusun membentuk angka 0 sampai 9. Pada metode ini, digunakan tujuh buah garis atau segmen yang tersusun sehungga dengan manyalakan garis-garis tertentu akan membentuk angka decimal, umumnya disebut tampilan 7-segmen.



Gambar 2.7 Simbol 7-Segmen dan Komponen 7-Segmen

2.3.5 Matrik Keypad

Keypad adalah istilah umum untuk elemen input dari PC, tetapi juga istilah ini dapat diaplikasikan pada sistem MCU, tentunya dengan jumlah saklar yang lebih sedikit dibanding dengan PC. Jika jumlahnya agak banyak, misalnya lebih dari 10, maka jika dihubungkan langsung ke port akan mengurangi penggunaan I/O untuk keperluan lain. Cara yang umum adalah dengan matrik. Cara ini menghemat jumlah I/O, karena dengan 16 saklar, hanya memerlukan 8 I/O saja. Akan tetapi cara ini akan membuat program pendeteksian saklar yang relatif lebih rumit.



Gambar 2.8 Matrik Keypad dan Circuit Diagram

Pembacaan status saklar cara matrik adalah sebagai berikut:

- Jika kolom 1 = 0, kemudian

  Baris 1 = 0, maka tombol 1 ditekan

  Baris 2 = 0, maka tombol 4 ditekan

  Baris 3 = 0, maka tombol 7 ditekan

  Baris 4 = 0, maka tombol * ditekan
- Jika kolom 2 = 0, kemudian

  Baris 1 = 0, maka tombol 2 ditekan

  Baris 2 = 0, maka tombol 5 ditekan

  Baris 3 = 0, maka tombol 8 ditekan

  Baris 4 = 0, maka tombol 0 ditekan
- Jika kolom 3 = 0, kemudian

  Baris 1 = 0, maka tombol 3 ditekan

  Baris 2 = 0, maka tombol 6 ditekan

  Baris 3 = 0, maka tombol 9 ditekan

  Baris 4 = 0, maka tombol # ditekan