

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Karya ilmiah yang penulis buat dalam tugas akhir ini sama – sama membuat sistem pengontrolan pintu otomatis. Dalam karya ilmiah sebelumnya membahas tentang perencanaan pengontrolan pintu otomatis.

2.2 Teori Dasar

2.2.1 *Microcontroller AT 89C51*

Penggunaan *microcontroller AT89C51* akan mempermudah dalam perancangan sebuah sistem karena telah dilengkapi dengan *Flash Perom* yang akan ditambah dengan komponen – komponen lain sebagai pendukung. Keuntungan menggunakan *microcontroller AT 89C51* antara lain : lebih ekonomis dan menghemat biaya karena program yang dimasukkan ke dalam chip *microcontroller AT 89C51* ini hampir sama dengan yang digunakan untuk menyimpan *BIOS* dalam sebuah komputer. Selain itu juga dapat menghemat waktu perencanaan dan proses pembuatan yang relatif singkat karena jumlah komponen yang digunakan lebih sedikit.

Microcontroller AT 89C51 memiliki *flash memory* (memori program internal) dengan *single IC 40 pin* yang memudahkan dalam menghapus dan memprogram kembali berulang kali. Oleh karena itu banyak industri yang

menggunakan *Microcontroller AT 89C51* ini sebagai pengontrol sistem.

Spesifikasi *microcontroller AT 89C51* adalah sebagai berikut :

1. *Multiplex* mode dan kemampuan pemrograman tingkat tinggi untuk *serial port*.
2. Memiliki jalur input 128 dan dua arah output yang dikelompokkan menjadi 4 *byte port*.
3. Dua buah *counter* 16 bit,
4. *Full duplex Serial Port*.
5. Empat K (4096) bytes dalam sistem *Reprogrammable Flash Memory*.
6. *Memory data* eksternal yang mencapai 64 Kb.

Unit mikrokontroler ini merupakan otak dari keseluruhan sistem yang direncanakan. Dimana mikrokontroler ini akan mendeteksi nomor masukan . Informasi tersebut akan diolah dengan mikrokontroler, sehingga mikrokontroler akan membangkitkan sinyal informasi yang nantinya digunakan untuk pengaturan gerak suatu motor DC. Dimana motor DC tersebut difungsikan untuk membuka dan menutup dari posisi pintu.

Berikut adalah gambar dari *Microcontroller AT89C51*:

P1.0	1	40	VCC
P1.1	2	39	P0.0
P1.2	3	38	P0.1
P1.3	4	37	P0.2
P1.4	5	36	P0.3
P1.5	6	35	P0.4
P1.6	7	34	P0.5
P1.7	8	33	P0.6
RST	9	32	P0.7
RXD(P3.0)	10	31	EA/VPP
TXD(P3.1)	11	30	ALE
INT(P3.2)	12	29	PSEN
INT1(P3.3)	13	28	P2.7
T0(P3.4)	14	27	P2.7
T1(P3.5)	15	26	P2.7
WR(P3.6)	16	25	P2.7
RD(P3.7)	17	24	P2.7
XTAL2	18	23	P2.7
XTAL1	19	22	P2.7
GND	20	21	P2.7

Gambar 2.1

Konfigurasi pin 8951 (Microcontroller AT89C51)

Konfigurasi dari kaki – kaki tersebut adalah sebagai berikut :

1. VSS

Dihubungkan dengan ground rangkaian.

2. VCC

Dihubungkan dengan sumber tegangan.

3. Port 0 (P0.0 – P0.7)

Port 0 merupakan I / O 8 bit dua arah. Port ini digunakan sebagai multipleks bus alamat rendah dan bus data selama pengaksesan ke memory luar.

4. Port 1 (P1.0 – P1.7)

Port 1 dapat berfungsi sebagai input atau output dan bekerja baik untuk operasi bit maupun byte tergantung dari pengaturan software.

5. Port 2 (P2.0 – P2.7)

Port 2 dapat digunakan sebagai alamat bus byte tinggi selama adanya akses ke memory program luar.

6. Port 3 (P3.0 – P3.7)

Port ini selain mempunyai fungsi sebagai I / O juga mempunyai fungsi sebagai khusus sebagai berikut ;

- RD (P3.7) : sinyal pembacaan memory data luar.
- WR (P3.6) : sinyal penulisan memory data luar.
- T1 (P3.5) : masukan dari pewaktu / pencacah 1
- T0 (P3.4) : masukan dari pewaktu / pencacah 0
- INT 1 (P3.3) : masukan interupsi 1
- INT (P3.2) : masukan interupsi 2
- TXD (P3.1) : keluaran pengiriman data untuk serial port (asinkron) atau sebagai keluaran clock (sinkron)
- RXD (P3.0) : keluaran pengiriman data untuk serial (asinkron) atau sebagai masukan / keluaran data (sinkron).

7. RST / VPD

Merupakan pin input yang aktif tinggi, jika pin ini aktif tinggi selama dua siklus mesin ketika osilator bekerja akan mereset peralatan.

8. ALE (Address Latch Enables) / Prog (Pulse Program)

Pin ALE aktif tinggi mengeluarkan pulsa output untuk me-latch satu byte alamat rendah selama mengakses ke memory luar. ALE dapat mengendalikan 8 beban TTL. Pin ini merupakan input pulsa program yang aktif rendah selama pemrograman EPROM.

9. PSEN (Program Strobe Enable)

Pin ini aktif rendah yang merupakan Strobe pembacaan ke program memory eksternal.

10. XTAL 1 DAN XTAL 2

Pin XTAL 1 merupakan pin input ke penguat osilator pembalik dan pin XTAL 2 merupakan pin out dari penguat osilator pembalik.

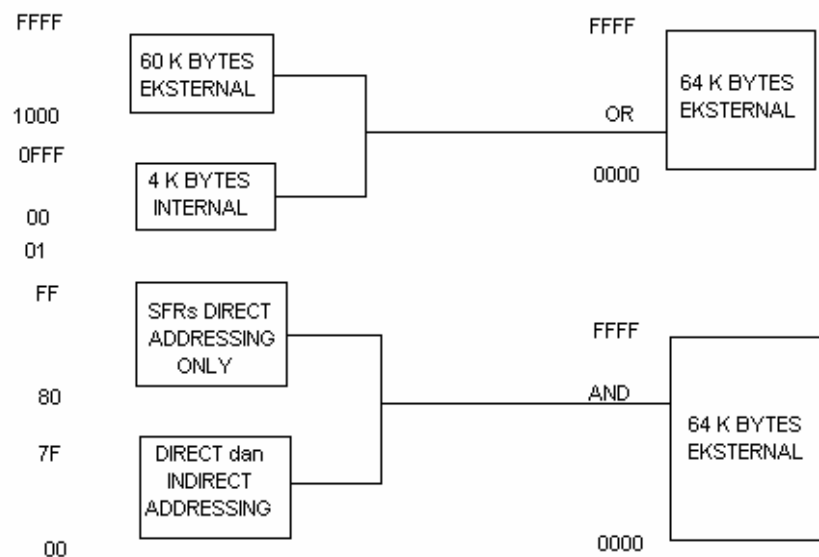
11. EA / VVP (Eksternal Access / Programming Supply Voltage)

Pin Ea harus dihold rendah secara eksternal atau dihubungkan ke ground agar 8951 dapat mengakses kode mesin dari program memory eksternal dengan lokasi \$0000 sampai \$0FFF.

2.2.2 Memory MC 8951

Pada microcontroller keluarga MCS 51, memiliki memory program dan memory data yang terpisah seperti ditunjukkan pada gambar 1.1.

Pemisahan dilakukan secara logika, sehingga CPU dapat mengakses sampai 64 K byte memory program dari 64 K byte memory data. Lebar memory data internal adalah 8 bit dan 16 bit (register PC dan register DPTR).



Gambar 2.2

Memory Program 8951

2.2.3 Memory Program

Memory Program merupakan tempat penyimpanan data yang permanen. Memory Program merupakan memory yang hanya dapat dibaca atau lebih dikenal dengan nama Read Only Memory (ROM). Data dalam ROM tidak akan terhapus meskipun catu daya dimatikan (bersifat non-volatile). Oleh karena itu ROM hanya dapat digunakan untuk menyimpam program.

Ada beberapa tipe ROM, diantaranya ROM murni, yaitu memory yang sudah diprogram oleh pabrik, PROM, EPROM, dan EEPROM. PROM merupakan memory yang dapat diprogram oleh pemakai tetapi tidak dapat diprogram ulang. EPROM merupakan PROM yang dapat diprogram ulang. ROM inilah yang terdapat pada microcontroller 8751, hal ini ditandai dengan adanya kaca pada konstruksi IC 8751 yang digunakan untuk menghapus atau memprogram ulang program yang sudah ada.

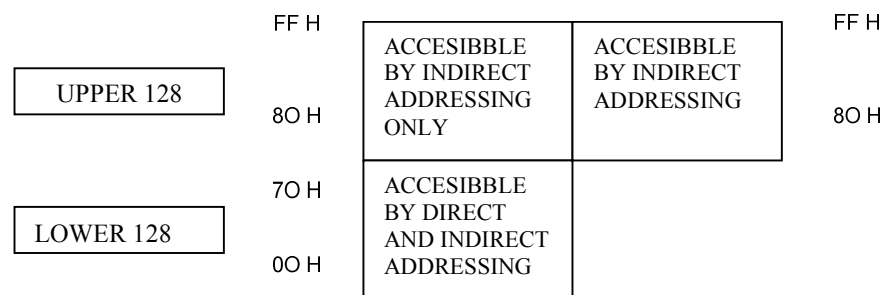
EEPROM pada prinsipnya sama dengan EPROM perbedaannya terletak pada pengisian dan penghapusan program. Pada EPROM untuk mengisi ulang program harus dihapus dulu menggunakan ultra violet, sedangkan pada EEPROM pengisian dapat dilakukan secara langsung, yaitu dengan menumpuk program yang lama. Sehingga penggunaan EEPROM lebih fleksibel dan harganya lebih murah dibandingkan EPROM.

2.2.4 Memory Data

Memory data adalah tempat untuk menyimpan data yang sifatnya sementara, sehingga pada memory data bersifat volatile (data akan hilang jika tidak dicatu). Memory data lebih dikenal dengan RAM (Random Acces Memory) yaitu dapat dilakukan pembacaan dan penulisan data pada alamat yang tersedia.

Selain pada eksternal, MCS51 Juga mempunyai 128 byte internal ditambah dengan sejumlah register fungsi khusus atau *Spesial Funtion Register* (SFRs).

Pada keluarga mikrokontroler 8951, ruang data memory internal terbagi menjadi 3 blok yang disebut sebagai *lower* 128, *upper* 128 dan ruang SRFs sebagaimana tertera pada gambar dibawah ini :



Gambar 2.3

Internal Memori Data

Pada 128 byte atas (*upper*) ditempati suatu fungsi register yang mempunyai fungsi khusus yang disebut *Special Function Register* (SFRs), Ruang dari register fungsi khusus ini adalah 80H sampai FFh .Berikut ini adalah contoh dari *vector* alamat pada *Special Function Register* :

1. *Accumulator* (Acc) atau register A dan B

Kedua register ini digunakan untuk operasi perkalian dan pembagian

2. *Program Status Word*

Register ini meliputi bit-bit:CY (*carry*),AC (*auxiliary carry*) ,FO (sebagai

Flag), RSO dan RSI (untuk pemilihan register bank), OV (*Over Flow*), Parity (*Parity Flag*).

3. *Stack Pointer*

SP merupakan register yang digunakan untuk petunjuk alamat. Register ini berguna apabila suatu routine pada program utama.

4. *Data Pointer High (DPH) dan Data Pointer Low (DPL)*

DPTR adalah suatu register yang digunakan untuk pengalamatan tidak langsung. Register ini selain digunakan untuk mengakses memori program baik eksternal maupun internal, juga digunakan untuk alamat external data. DPTR ini dikontrol oleh dua buah register 8 bit yaitu DPL dan DPH.

5. *Port 0, Port 1, Port 2, Port 3*

Pada keluarga 8051 masing-masing port dapat dapat dialamati baik secara byte maupun bit. masing-masing port merupakan port bi-directional yang artinya dapat berjalan dua arah (*input dan output*). Port 0 dan port 2 digunakan untuk pengalamatan dari luar, port 1 digunakan untuk I/O, sedang port 3 berisi sinyal-sinyal control seperti interrupt serial, WR dan RD.

6. *Register Prioritas Interupt*

Merupakan register yang berisi bit-bit untuk mengaktifkan prioritas dari suatu

interupsi yang ada pada mikrokontroler pada tahapan yang diinginkan

7. *Interrupt Enable Register*

EI merupakan register yang berisi bit-bit untuk menghidupkan atau mematikan sumber interrupt

8. *Timer./ Counter Control Register*

TCON merupakan suatu register yang berisi bit-bit untuk memulai / menghentikasn pencacah/ pewaktu

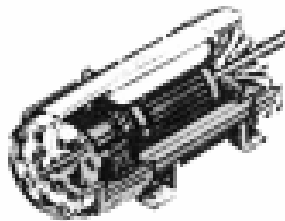
9. *Serial Control Buffer(SBUF)*

Register ini digunakan untuk menampung data dari masukan (*SBUF IN*) atau keluaran (*SBUF OUT*) dari serial port

2.3 Motor DC

Untuk dapat membuka dan menutup pintu dalam tugas akhir ini menggunakan motor DC. Motor DC merupakan mesin yang berfungsi sebagai pengubah tenaga listrik searah menjadi tenaga gerak atau mekanik berupa putaran motor. Motor DC sendiri terdiri dari bagian yang diam yaitu stator dan bagian yang gerak atau berputar disebut rotor. Mesin DC banyak digunakan dalam industri modern, alasannya adalah karena kinerja motor DC lebih cepat dan lebih mudah diatur.

Pada tugas akhir ini digunakan motor DC sebagai penggerak pintu dengan memanfaatkan putaran dari motor DC tersebut. Agar arah putaran dari motor dapat disesuaikan dengan gerakan dari daun pintu maka motor DC ini perlu dilengkapi dengan belt dan gear-gear mekanik.



Gambar 2.4

Motor DC (General Electric Company)

2.4 Bahasa Pemrograman Mikrokontroler

Sebuah mikrokontroler tidak akan bekerja jika tidak ada program didalamnya, program tersebut memberitahu mikrokontroler apa yang harus dilakukan. Setiap instruksi software untuk mikrokontroler tidak ada yang sama. Sebuah mikrokontroler tidak dapat memahami instruksi – instruksi yang berlaku pada mikrokontroler lain. Instruksi – instruksi inilah yang dikenal sebagai bahasa pemrograman mikrokontroler.

2.4.1 Simbol Assembler khusus

Assembler telah menyediakan beberapa simbol untuk menunjukkan register tertentu sebagai operand. Tabel berikut menunjukkan simbol

assembler tersebut.

No	Simbol Khusus	Arti
1	A	Akumulator
2	R0 ... R7	Register serbaguna
3	DPTR	Data pointer, Register 16 bit
4	PC	Program counter. Register 16 bit yang berisi alamat instruksi berikutnya yang akan dijalankan
5	C	Carry flag
6	AB	Akumulator / register B, pasangan register untuk perkalian dan pembagian

Gambar 2.5

Tabel Assembler Khusus

2.4.1.1 Pengalamatan Tak Langsung

Pengalamatan tak langsung menunjukkan ke sebuah register yang berisi lokasi alamat memori yang akan digunakan dalam operasi. Lokasi yang nyata tergantung pada isi register saat instruksi dijalankan. Untuk melaksanakan pengalamatan tak langsung digunakan simbol @.

ADD A,@R0 Tambahkan isi RAM yang lokasinya ditunjukkan oleh *register* R0 ke akumulator.

- DEC @R1 kurangi satu isi RAM yang alamatnya ditunjukkan oleh *register* R1.
- MOVX @DPTR,A Pindahkan isi dari akumulator ke memori luar yang lokasinya ditunjukkan oleh data pointer (DPTR).

2.4.1.2 Pengalamatan Langsung

Pengalamatan langsung dilakukan dengan memberi nilai ke suatu register secara langsung. Untuk melaksanakan hal tersebut digunakan tanda #.

- MOV A, #01H Isi akumulator dengan bilangan 01H.
- MOV DPTR, #19AB Isi register DPTR dengan bilangan 19ABH.
- MOV P3,A Pindahkan isi akumulator ke alamat data B0H (B0H adalah alamat port 3).
- INC 50 Naikan lokasi 50 (desimal) dalam memory.

2.4.1.3 Pengalamatan Bit

Pengalamatan bit Adalah penunjukkan alamat lokasi bit baik dalam RAM internal (*byte* 32-47) atau bit perangkat keras. Untuk melakukan pengalamatan bit digunakan simbol titik (.), misalnya FLAGS.3.40.5.21H.5 dan ACC.7.

2.4.1.4 Pengalamatan Kode

Ada tiga macam yang digunakan dalam mengalamatkan kode, yaitu *relative jump*, *in-block jump* atau *call*, dan *long jump* atau *call*.

2.4.1.5 Perangkat Instruksi

Mikrokontroler Intel 89C51 memiliki 256 perangkat instruksi. Seluruh instruksi dapat dikelompokkan dalam 4 bagian yang meliputi instruksi 1 *byte* sampai 4 *byte*.

Apabila frekuensi *clock* mikrokontroler yang digunakan adalah 12 Mhz. Kecepatan pelaksanaan instruksi akan bervariasi dari 1 hingga 4 mikro detik. Perangkat instruksi mikrokontroler 89C51 dapat dibagi menjadi 5 kelompok sebagai berikut :

- Instruksi Transfer Data

Instruksi ini memindahkan data antara register-register, memori-memori, register-memori, antar muka-register, dan antar muka-memori.

- Instruksi Aritmatika

Instruksi ini melaksanakan operasi aritmatika yang meliputi penjumlahan, pengurangan, penambahan satu (increment), pengurangan satu (decrement), perkalian dan pembagian.

- Instruksi Logika dan Manipulasi Bit

Melaksanakan operasi logika AND, OR, XOR, perbandingan, penggeseran, dan komplemen data.

- Instruksi Percabangan

Instruksi ini mengubah urutan normal pelaksanaan suatu program. Dengan instruksi ini program yang dilaksanakan akan mencabang ke suatu alamat tertentu. Instruksi percabangan dibedakan atas percabangan bersyarat dan tanpa syarat.

- Instruksi Stack, I/O, dan Kontrol

Instruksi ini mengatur penggunaan stack, membaca / menulis port I/O, serta pengontrolan- pengontrolan.

Perangkat instruksi mikrokontroler 89C51 secara lengkap adalah sebagai berikut :

ACALL

Absolute Call

Mnemonic : ACALL

Operand : Alamat kode

Format : ACALL alamat kode

Keterangan instruksi ini menyimpan isi yang telah dimatikan dari pencacah program pada stack. Byte bawah dari pencacah program selalu ditempatkan pada stack pertama.

ADD

Add Immediate Data

Mnemonic : ADD

Operand : A Akumulator

: data-256 <= data <= +256

Format : ADD A, # data

Keterangan instruksi ini menambah 8 bit data langsung ke dalam isi akumulator dan menyimpan hasilnya pada akumulator.

ADD

Add Indirect data

Mnemonic : ADD

Operand : A Akumulator
: Rr Register $0 \leq r \leq 1$

Format : ADD A, @Rr

Keterangan instruksi ini menambahkan isi data memori yang lokasinya ditunjukkan oleh nilai register r ke isi akumulator dan menyimpan hasilnya dalam akumulator.

ADD

Add Register

Mnemonic : ADD

Operand : A Akumulator
: Rr Register 0 sampai 7

Format : ADD A, Rr

Keterangan instruksi ini menambahkan isi register r dengan isi akumulator dan menyimpan hasilnya dalam akumulator.

ADD

Add Memori

Mnemonic : ADD

Operand : A Akumulator
: Alamat data $0 \leq \text{alamat data} = 256$

Format : ADD A, alamat data

Keterangan instruksi ini menambahkan isi alamat data ke isi akumulator dan menyimpan isinya dalam akumulator.

ADDC

Addc Carry Plus Immediate Data to Accumulator

Mnemonic : ADDC

Operand : A Akumulator

: data -256 <= data <= +256

Format : ADDC A, # data

Keterangan instruksi ini menambah isi carry flag (0 atau 1) kedalam isi akumulator.

Data langsung 8 bit ditambahkan dalam akumulator.

ADDC

Addc Carry Plus Indirect Address to Accumulator

Mnemonic : ADDC

Operand : A Akumulator

: Rr Register 0 <= r <= 1

Format : ADDC A, @Rr

Keterangan instruksi ini menambah isi carry flag (0 atau 1) dengan isi akumulator. Isi

data memori pada lokasi yang ditunjukkan oleh register Rr ditambahkan dan hasilnya

disimpan di akumulator.

ADDC

Addc Carry Plus Register to Accumulator

Mnemonic : ADDC

Operand : A Akumulator

: Rr Register 0 <= r <= 7

Format : ADDC A, Rr

Keterangan instruksi ini menambah isi carry flag dengan isi akumulator. Isi register r ditambahkan dan hasilnya disimpan di akumulator.

ADDC

Add Carry Plus Indirect Address to Accumulator

Mnemonic : ADDC

Operand : A Akumulator

: Alamat Data 0 <= Alamat data <= 255

Format : ADDC A, alamat data

Keterangan instruksi ini menambah isi carry flag dengan isi akumulator. Isi dari alamat data tertentu ditambahkan pula. Hasilnya disimpan dalam akumulator.

AJMP

Absolute Jump Within 2K Byte Page

Mnemonic : AJMP

Operand : Alamat kode

Format : AJMP Alamat kode

Keterangan instruksi ini meletakkan bagian bawah 11 bit dari pencacah program dengan 11 bit alamat yang dikodekan.

ANL

Logical AND Indirect Address to Accumulator

Mnemonic : ANL

Operand : A Akumulator

: R Register $0 \leq r \leq 1$

Format : ANL A, @Rr

Keterangan instruksi ini meng-AND-kan isi memori yang lokasinya ditunjukkan oleh register r dengan isi akumulator.

ANL

Logical AND Immediate Data to Accumulator

Mnemonic : ANL

Operand : A Akumulator

: data $-256 \leq \text{data} \leq +256$

Format : ANL A, # data

Keterangan instruksi ini meng-AND-kan data 8 bit secara langsung dengan isi akumulator.

ANL

Logical AND Register to Accumulator

Mnemonic : ANL

Operand : A Akumulator

: R Register $0 \leq Rr \leq 7$

Format : ANL A, Rr

Keterangan instruksi ini meng-AND-kan isi register r dengan isi akumulator.

ANL

Logical AND Memori to Accumulator

Mnemonic : ANL

Operand : A Akumulator
: Alamat Data 0 <= Alamat Data <= 255

Format : ANL A, Alamat Data

Keterangan instruksi ini meng-AND-kan isi alamat tertentu dengan isi akumulator.

ANL

Logical AND Bit to Carry Flag

Mnemonic : ANL

Operand : C Carry Flag
: Alamat bit 0 <= Alamat bit <= 255

Format : ANL C, Alamat bit

Keterangan instruksi ini meng-AND-kan isi alamat tertentu dengan isi carry flag. Jika keduanya 1 maka hasilnya 1, selain itu hasilnya 0. hasilnya ditempatkan pada carry flag.