BAB I

TUGAS MATA KULIAH
SISTEM MIKROPROSESOR
DOSEN PEMBERI TUGAS : FATAH YASIN, ST, MT.

A. Deskripsi Tugas

1. Jelaskan perbedaan mikroprosesor dan mikrokontroler.
2. Jelaskan mode-mode pengalamatan yang ada pada salah satu mikrokontroler yang anda ketahui.
3. Apa yang dimaksud dengan;
    a. Address bus, data bus dan control bus.
    b. Accumulator.
    c. Register.
    d. Baud rate.
4. Buatkan program dengan bahasa assembly untuk menjumlahkan angka: 4+5+6 dan hasilnya disimpan di alamat memori 80H.

B. Penyelesaian

1. Jelaskan Perbedaan Mikroprosesor dan Mikrokontroler

1.1 Mikroprosesor

Mikroposessor juga dikenal dengan istilah *central processing unit* adalah bagian utama dari sistem mikrokomputer. Blok diagram dari CPU yang terdiri dari *arithmetic logic unit (ALU), program counter (PC), stack pointer (SP),* beberapa register kerja, rangkaian clock dan rangkaian interupsi. Untuk membuat sistem mikrokomputer lengkap harus ditambahkan memori program (ROM), memori data (RAM), dekoder memori, osilator dan beberapa peralatan input/output. Kunci dari perencanaan mikroprosessor adalah sifatnya yang serbaguna yaitu perencanaan sistem mikroprosesor dapat dari kecil sampai sangat besar tergantung pada kebutuhan pemakaian.

1.2 Mikrokontroler

Mikrokontroler merupakan sebuah prosesor yang dilengkapi dengan memori (RAM/ROM) dan fasilitas I/O internal dikemas dalam satu chip tunggal. Mikrokontroller direncanakan khusus untuk kontroler digital fungsi tertentu. Untuk mengaplikasikan mikrokontroler tidak diperlukan lagi unit-unit tambahan seperti halnya pada rancangan menggunakan mikroprosesor. Pada mikrokontroler (MC) memori program, memori kerja dan fungsi-fungsi lainnya

(peripheral) terdapat pada satu chip, sedangkan di mikroprosesor (MP) harus ditambahkan peripheralnya.

2. Jelaskan mode-mode pengalamatan yang ada pada salah satu Mikrokontroler yang Anda ketahui ?

Mode-mode pengalamatan yang ada pada mikrokontroler 8031

Karena arsitektur 8031 membedakan antara memori data dengan program memori, maka masing-masing mempunyai mode-mode pengalamatan yang berbeda-beda. Mode pengalamatan adalah cara memindahkan data dari alamat asal menuju alamat yang dituliskan dalam kode mnemonic. Ada lima metode pengalamatan *source operand*, yaitu:

a. *Register addressing*
b. *Direct addressing*
c. *Indirect Addressing*
d. *Immediate Addressing*
e. *Base-register-plus index-register-indirect addressing*

Tiga mode yang pertama juga dapat berfungsi untuk alamat suatu *destination operand*.

Pada umumnya instruksi-instruksi mempunyai *destination source* yang menspesifikasikan tipe data, mode pengalamatan dan *operand-operand* yang diperlukan. Selain untuk instruksi MOV, *destination operand* juga berfungsi sebagai *source operand* yang berlokasi di memori data internal. Pemilihan antara memori program dan memori data eksternal

ditentukan oleh *mnemonic*, kecuali untuk *immediate operand*. Berikut ini penjelasan dari mode-mode pengalamatan yang telah disebut diatas.

2.1 Register addressing

Pada mode *register addressing* maka nama suatu register digunakan sebagai satu kesatuan dari *opcode* baik sebagai data asal atau data tujuan. Untuk register A,DPTR dan R0-R7 dapat digunakan langsung sebagai bagian dari intruksi sedang untuk register lain harus digunakan perintah *psudo 'equate'* dahulu yang merupakan perintah dalam bahasa *assembler* (bahasa mesin). Format dari intruksi *register addressing* adalah sebagai berikut:

Opcode (Ri)	R0 - R7
Intruksi	Data

Contoh:

```
MOV   A,R0    ; Mengkopi data dari register R0 ke Accumulator

MOV   R5,A    ; Mengkopi data dari Accumulator ke register R5
```

2.2 *Direct addressing* (Pengalamatan Langsung)

Pada mode pengalamatan langsung, tempat data yang akan diproses alamatnya ditunjukkan langsung dalam perintah. Selain lokasi memori, register fungsi khusus (SFR) juga dapat dialamati secara langsung menggunakan lokasi RAM internal yang sesuai.

Bentuk format perintah mode *Direct Addressing* adalah sebagai berikut:

Opcode (Add)	Alamat RAM
Intruksi	Data

Contoh :

```
MOV   A,80H      ; Mengkopi data dari pin-pin port 0 ke register A
MOV   80H,A      ; Mengkopi data dari register A ke pin-pin port 0
MOV   R0,12H     ; Mengkopi data dari alamat RAM 12H ke R0
MOV   3A,#3AH    ; Mengkopi data 3AH ke lokasi RAM 3A
```

2.3 *Indirect Addressing* (Pengalamatan Tak Langsung)

Pada pengalamatan tidak langsung data yang akan diproses alamat sebenarnya tidak ditunjukkan secara langsung tetapi disimpan pada salah satu register (R0/R1). Pengalamatan tidak langsung pada kode mnemonicnya menggunakan simbol @. Format dari pengalamatan tidak langsung adalah sebagai berikut:

Opcode (@Rn)	R0 atau R1	Alamat RAM
Intruksi	Alamat	Data

Contoh:

```
MOV   @R1,A      ; Mengkopi isi A ke alamat di dalam R1
MOV   @R1,35H    ; Mengkopi angka 35H ke alamat di dalam R1
MOV   @R0,80H    ; Data pada pin-pin port 0 dikopi ke alamat
                   dalam R1
```

```
MOV   address,@R0   ; Mengkopi isi dari alamat di dalam R0
                      ke alamat
MOV   A,@R0         ; Mengkopi isi dan alamat di dalam R0
                      ke  register A
```

2.4 Immediate Addressing

Cara yang paling sederhana untuk memindahkan data menuju alamat tujuan adalah mode *immediate*. Pada mode ini data merupakan satu kesatuan dalam intruksi itu sendiri. Bila 8031 melaksanakan perintah mode *immediate* maka secara otomatis Program Counter akan bertambah menunjuk byte setelah *byte opcode*. Perintah dalam kode mnemonic untuk mode *immediate* menggunakan tanda (#) di depan datanya. Diagram dari mode *immediate* adalah sebagai berikut :

Opcode (#n)	Byte berikutnya
Intruksi	Data

Contoh:

```
MOV   A,#0FFH          ;Mengkopi data FFH ke Accumulator
MOV   R4,#1AH          ; Mengkopi data 1AH ke register R4
MOV   DPTR,#0FABCH     ; Mengkopi data FABCH ke DPTR
MOV   R5,A             ;Mengkopi data dari register A ke R5
```

Dalam contoh-contoh tersebut terdapat *operand* sebelah kiri sebagai tujuan dan *operand* sebelah kanan sebagai sumber/asal data. Misalnya untuk MOV A,#0FFH, maka A merupakan alamat tujuan dalam hal ini *Accumulator* dan #0FFH merupakan data yang akan dipindahkan.

2.5 Base-register-plus index-register-indirect addressing

Contoh:

MOVC A,@A+DPTR	; Mengkopi *code byte*, yang terdapat pada alamat ROM yang lokasinya ditunjukkan oleh penjumlahan A dan DPTR ke register A
MOVC A,@A+PC	; Mengkopi *code byte*, yang terdapat pada alamat ROM yang lokasinya ditunjukkan oleh penjumlahan A dan *Program Counter* (PC) ke register A

3. Apa yang dimaksud dengan ;

a. *Address bus* atau bus alamat adalah kumpulan jalur yang membawa informasi tentang lokasi pengingat yang dituju tempat data akan dikirimkan atau diterima.

b. *Data bus* atau bus data adalah jalur yang berfungsi untuk menyalurkan data dari suatu bagian ke bagian lainnya. Berisi 8, 16, 32 jalur sinyal paralel atau lebih. Jalur-jalur data adalah dua arah (*bidirectional*). CPU dapat membaca dan mengirim data dari/ke memori atau *port*. Banyak perangkat pada sistem yang dicantolkan ke bus data tapi hanya satu perangkat pada satu saat yang dapat memakainya. Untuk mengatur ini, perangkat harus mempunyai tiga *state (tristate*) agar dapat dipasang pada bus data.

c. *Control bus* atau bus kendali adalah sekumpulan jalur sinyal yang digunakan untuk mengontrol pemindahan data di seluruh sistem dan juga untuk berinteraksi dengan perangkat-perangkat eksternal.

d. *Accumulator* (ACC) adalah register akumulator. *Mnemonik* untuk intruksi spesifik akumulator ini secara sederhana dapat disingkat A.

e. Register adalah Memori yang sangat cepat dalam transfer datanya, bertugas membantu operasi yang dilakukan pemroses, terutama sebagai tempat operan-operan. Register ini dikategorikan menjadi dua, yaitu (1) register yang terlihat pemakai, seperti data register dan *address register*. (2) register untuk kendali status.

Dilambangkan dengan (R), *registered trademark.* Sudah terdaftar secara hukum. Disingkat dengan (R). Memori yang mencatat data dan dapat melakukan proses data dengan kecepatan tinggi.

f. Baud rate adalah jumlah data yang dapat ditransfer melalui sebuah *interface serial*.

4. Buatkan program dengan bahasa assembly untuk menjumlahkan angka: 4+5+6 dan hasilnya disimpan di alamat memori 80H ?

Program Assembly untuk penjumlahan

```
Mov   ax, 4          ; memindahkan 4 ke dalam register ax
Add   ax, 5          ; menambahkan nilai 5h terhadap register ax
Add   ax, 6          ; menambahkan nilai 6h terhadap register ax
Mov   [080], ax      ; menyimpan hasil jumlah ax dalam lokasi 80h
```