



Nama	Irfan Arif Maulana
NPM	1906379270

Kode Asisten	AS
Modul 2	Pemindahan Data, Aritmatika, dan Logika

PART I – Analisis Program

Perhatikan program berikut untuk menjawab pertanyaan-pertanyaan di bawahnya!

```
.model small

.data

    DATA1 db 15h
    DATA2 db 10h

.code

.startup

    mov AL, DATA1

    mov BX, offset DATA2

    mov DX, [BX]

    mul DL

    mov 1234h, AX

    and AX, 0h

    mov AL, DATA1

    div DL

    mov DX, AX

    mov 1238h, AX

.exit

END
```

1. Apa fungsi dari instruksi MOV? Bagaimana cara kerjanya?

Fungsi dari instruksi MOV adalah untuk menyalin atau meng-*copy* sebuah data dari sumber ke tujuan atau dari *source* ke *destination*. Oleh karena fungsinya adalah menyalin, maka data yang ada di sumber akan tetap tersimpan.

Sintaks perintah MOV adalah sebagai berikut: MOV *destination*, *source*

Di mana *source* merupakan register asal yang *value*-nya hendak disalin dan *destination* merupakan register atau memori tujuan yang hendak diberikan *value*.

2. Apa fungsi dari instruksi MUL? Mengapa hasil dari perkalian DATA1 dan DATA2 adalah 01h dan 50h?

Fungsi dari instruksi MUL adalah untuk melakukan operasi aritmatika perkalian. Di mana *multipland* dari perkalian tersebut merupakan register AX atau AL (bergantung dengan besar *multiplier*-nya). Sedangkan *multiplier* dari perkaliannya bisa register apa pun. Hasil perkalian tersebut disimpan pada register AX. Sintaks dari MUL adalah sebagai berikut: MUL *multiplier*.

Perkalian antara DATA1 dan DATA2 yang mana keduanya merupakan variabel berukuran 2 bit hexa akan menghasilkan setidaknya 3 bit hexa. Di mana pada kasus ini $15H \times 10H = 150H$. Oleh karena masing-masing dari AH dan AL yang mana keduanya merupakan tempat penyimpanan hasil perkalian, hanya dapat memuat 2 bit hexa, maka terlihat hasil perkaliannya berupa 01H dan 50H. Dengan 01H merupakan konten dari AH sedangkan 50H merupakan konten dari AL.

3. Apa fungsi dari instruksi DIV? Mengapa hasil dari pembagian DATA1 dan DATA2 adalah 05h dan 01h?

Fungsi dari instruksi DIV adalah untuk melakukan operasi aritmatika pembagian. *Dividend* dari DIV adalah register AX dan *divisor*-nya dapat berupa register apa saja. Sintaks DIV adalah: DIV *divisor*. Hasil dari pembagian ini disimpan pada register AX dengan AH menyimpan sisa pembagian (*remainder*) dan AL menyimpan hasil pembagian (*quotient*).

Pembagian antara DATA1 dan DATA2 yang secara berurutan masing-masing nilainya adalah 15H dan 10H akan menghasilkan hasil pembagian 01H dengan sisa pembagian sebesar 05H. Oleh karena demikian, register AX yang fungsinya menyimpan hasil dan sisa pembagian tersebut akan tampak bernilai 0501H. Dengan 05H merupakan konten dari register AH sebagai sisa pembagian sedangkan 01H merupakan konten dari register AL sebagai hasil pembagian.

4. Sebutkan jenis addressing mana saja yang digunakan pada program di atas, dan dimana jenis tersebut digunakan!

Jenis *addressing* yang digunakan pada program di atas adalah antar register, *direct*, dan *indirect* dengan implementasinya sebagai berikut:

- Antar register : MOV DX, AX
- *Direct* : MOV AL, DATA1
: MOV 1234H, AX
: MOV 1238H, AX
- *Indirect* : MOV DX, [BX]

5. Apa yang dilakukan pada program di atas?

Tujuan utama dari program di atas adalah untuk mendemonstrasikan operasi aritmatika perkalian (MUL) dan pembagian (DIV). Dengan mengimplementasikan beberapa mode *data addressing* yang berbeda-beda, tepatnya antar register, *direct*, dan *indirect*. Diimplementasikan pula operasi logika untuk mengosongkan register.

Pertama-tama, program tersebut mendeklarasi dua buah variabel DATA1 dan DATA2. Kemudian DATA1 disalin nilainya ke register AL. Sedangkan DATA2 disalin alamatnya ke BX. Alamat DATA2 ini kemudian diambil *value*-nya melalui *indirect addressing* untuk disalin ke DX. Setelah nilai-nilai tersebut disalin, register AL dan DL siap untuk dilakukan perkalian. Hasil perkalian tersebut kemudian disimpan ke *data segment* 1234H.

Pada bagian kedua program, register AX dikosongkan terlebih dahulu dengan operasi logika guna menyiapkan operasi pembagian. Register AL diisi kembali dengan DATA1 dan dibagi dengan register DL yang dari awal sudah terisi DATA2. Dengan begitu pembagian dapat dilakukan. Hasil dan sisa pembagian ini kemudian disalin ke *data segment* 1238H.

PART II – Programming

1. Buatlah sebuah program yang menyimpan NPM kamu pada suatu variabel dengan keadaan dipisahkan setiap 2 digit (contoh: DATA3 db 18, 06, 20, 03, 81), kemudian menjumlahkan semua angka tersebut dan disimpan dalam lokasi memori 1234H!

Ketentuan:

- Program harus menggunakan minimal 3 jenis addressing yang berbeda

```

01 .model small
02
03 .data
04
05     DATA3 db 19H, 06H, 37H, 92H, 70H    ; 1906379270
06
07 .code
08
09 .startup
10
11     MOV AL, DATA3    ; Direct addressing mode
12
13     MOV BX, offset DATA3+1
14     MOV DX, [BX]      ; Indirect addressing mode
15
16     ADD AL, DH
17     ADD AL, DL
18
19     AND BX, 0H        ; Clear BX
20     MOV DX, 0H        ; Immediate addressing mode
21
22     MOV BL, DATA3+3
23     MOV DL, DATA3+4
24
25     ADD BX, DX
26     MOV DX, BX        ; Register addressing mode
27
28     ADD AX, DX
29
30     MOV [1234H], AX
31
32 .exit
33
34 END

```