
**IMPLEMENTASI PENGISIAN REGISTER DATA DAN OPERASI ARITMATIKA
MENGUNAKAN PROGRAM *COMMAND PROMPT*, BAHASA *ASSEMBLER*****Hambali**

Program Studi Sistem Informasi, STMIK Logika Medan

email: hambali.8645@yahoo.com

Abstract: Based on the type of information stored microprocessors can be divided into data registers, address registers, register flags and instruction registers. The filling of the application data register can be done by filling the register data AX, BX, CX, DX with the integration of 8 bit and 16 bit microprocessors with register AH, AL, BH, BL to DH register, DL using some programming language command assembler and application program run, cmd and commandprompt. Input data register on the use of data filling in data registers and address registers can be seen from some initial input registers are desired.

Keywords: data registers, assembler language

Abstrak: Berdasarkan jenis informasi yang disimpan mikroprosesor dapat dibagi menjadi register data, register alamat, register flag dan register instruksi. Pengisian register data penerapannya dapat dilakukan dengan dengan pengisian data register AX,BX,CX,DX dengan penggabungan mikroprosesor 8 bit dan 16 bit dengan register AH,AL,BH,BL sampai dengan register DH, DL menggunakan beberapa perintah bahasa pemrograman assembler dan aplikasi program run, cmd dan commandprompt. Inputan data register pada penggunaan pengisian data pada register data dan register alamat dapat dilihat dari beberapa inputan register awal yang diinginkan.

Kata kunci: register data, bahasa assembler

PENDAHULUAN

Pada bagan organisasi komputer, bagian memori diletakan terpisah dari mikroprosesor, jika bagan tersebut diimplementasikan, mikroprosesor harus mengakses memori setiap saat. Karena kecepatan memori jauh lebih lambat dari kecepatan mikroprosesor, maka kecepatan kerja mikroprosesor akan dipengaruhi oleh kecepatan memori.

Mikroprosesor dapat dikelompokkan menurut teknologi yang dipergunakan, menurut jumlah bit data, menurut struktur atau menurut kemampuan/karakteristik mikroprosesor dan menurut fungsi dari mikroprosesor itu sendiri. Berdasarkan jumlah bit data (*Word*

Size) pada waktu ini telah terdapat banyak macam mikroprosesor, mulai dari mikroprosesor 1bit, 4bit, 8 bit, 16 bit, 32 bit dan 64 bit.

Untuk mempercepat pemrosesan data data di dalam mikroprosesor, selain CU dan ALU, mikroprosesor juga akan membutuhkan memori yang memiliki kecepatan yang sama dengan mikroprosesor.

Untuk melakukannya, memori ini harus diimplementasikan di dalam mikroprosesor. Memori jenis ini disebut dengan register. Berdasarkan jenis informasi yang disimpan mikroprosesor dapat dibagi menjadi:

1. Register Data

Menyimpan data yang diperlukan untuk suatu operasi. Terdiri dari: (a) AX (*Accumulator*), (b) BX (*Base*), (c) CX (*Counter*), (d) DX (*Data*)

2. Register Alamat

Karena jumlah register data terbatas, maka sebagian besar data tetap diletakkan di memori. Untuk dapat mengakses memori, mikroprosesor membutuhkan alamat dari data tersebut pada memori. Alamat data tersebut di simpan di register yang disebut dengan register alamat. Terdiri dari: (a) SP (*Stack Pointer*), (b) BP (*Base Pointer*), (c) SI (*Source Index*), (d) DI (*Destination Index*), (e) DS (*Data Segment*) (f) ES (*Extra Segment*), (g) SS (*Stack Segment*), (h) CS (*Code Segment*) (i) IP (*Instruction Pointer*), dan (j) BX (*Base*) selain dapat digunakan sebagai register data dapat juga digunakan sebagai register alamat

3. Register Flag (Flags)

Digunakan untuk menyimpan status dari hasil operasi ALU. Terdiri dari: OF (*Overflow Flag*), DF, IF, TF, SF (*Sign Flag*), ZF (*Zero Flag*), AF, PF, CF (*Carry Flag*).

4. Register Instruksi

Digunakan untuk menyimpan instruksi yang sedang dikerjakan.

Untuk mempercepat pemrosesan data di dalam mikroprosesor, selain CU dan ALU, mikroprosesor juga akan membutuhkan memori yg memiliki kecepatan sama dengan uP. Untuk melakukannya, memori tersebut harus diimplementasikan didalam mikroprosesor. Memori ini disebut dengan register.

Operasi Penambahan (ADD)

Untuk menambah dalam bahasa assembler digunakan perintah **ADD** dan **ADC** serta **INC**. Perintah ADD digunakan dengan syntax : **ADD Tujuan,Asal**.

Perintah ADD ini akan menambahkan nilai pada Tujuan dan Asal. Hasil yang didapat akan ditaruh pada Tujuan, dalam bahasa pascal sama dengan instruksi **Tujuan:=Tujuan + Asal**.

Sebagai contohnya :

```
MOV AH,15h ; AH:=15h
MOV AL,4 ; AL:=4
```

ADD AH,AL ; AH:=AH+AL, jadi AH=19h

Operasi Pengurangan (SUB)

Untuk Operasi pengurangan dapat digunakan perintah SUB dengan syntax: **SUB Tujuan,Asal**

Perintah SUB akan mengurangkan nilai pada Tujuan dengan Asal. Hasil yang didapat akan ditaruh pada Tujuan, dalam bahasa pascal sama dengan instruksi

Tujuan:=Tujuan-Asal.

Contoh :

```
MOV AX,15 ; AX:=15
```

```
MOV BX,12 ; BX:=12
```

```
SUB AX,BX ; AX:=15-12=3
```

```
SUB AX,AX ; AX=0
```

Untuk menolak suatu register bisa anda kurangkan dengan dirinya sendiri seperti SUB AX,AX.

Operasi Perkalian

Untuk perkalian bisa digunakan perintah MUL dengan syntax: **MUL Sumber** Sumber disini dapat berupa suatu register 8 bit (Mis:BL,BH,...), register 16 bit (Mis: BX,DX,...) atau suatu variabel. Ada 2 kemungkinan yang akan terjadi pada perintah MUL ini sesuai dengan jenis perkalian 8 bit atau 16 bit.

Bila Sumber merupakan 8 bit seperti **MUL BH** maka komputer akan mengambil nilai yang terdapat pada BH dan nilai pada AL untuk dikalikan. Hasil yang didapat akan selalu disimpan pada register AX.

Bila sumber merupakan 16 bit seperti **MUL BX** maka komputer akan mengambil nilai yang terdapat pada BX dan nilai pada AX untuk dikalikan. Hasil yang didapat akan disimpan pada register DX dan AX (DX:AX), jadi register DX menyimpan *Word* tingginya dan AX menyimpan *Word* rendahnya.

Operasi Pembagian

Operasi pada pembagian pada dasarnya sama dengan perkalian. Untuk operasi pembagian digunakan perintah DIV dengan syntax: **DIV Sumber**

Bila **sumber** merupakan operand 8 bit seperti **DIV BH**, maka komputer akan mengambil nilai pada register AX dan

membaginya dengan nilai BH. Hasil pembagian 8 bit ini akan disimpan pada register AL dan sisa dari pembagian akan disimpan pada register AH.

METODOLOGI

Pengisian data dengan perintah MOV

Instruksi MOV

1. Register: Data (contoh: MOV AX,1234 :mengisi AX dgn data 1234)

2. Register: Register (contoh: MOV AX,BX: memindahkan isi BX ke AX)
3. Register: Memory (contoh: MOV AX,[BX] :memindahkan isi memori ke AX,dimana alamat dari datanya ada di BX)
4. Memory: Register (contoh: MOV [BX],AX : memindahkan isi AX ke memori, dimana datanya akan ditulis di memori pada alamat yg ada di BX)

Register Data

```
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0EA0 ES=0EA0 SS=0EA0 CS=0EA0 IP=0100 NV UP EI PL NZ NA PO NC
```

Gambar 1. Register Data

Register Alamat

```
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0EA0 ES=0EA0 SS=0EA0 CS=0EA0 IP=0100 NV UP EI PL NZ NA PO NC
```

Gambar 2. Register Alamat

Register Status (Flags)

Flag:

-	-	-	-	OF	DF	IF	TF	SF	ZF	-	AF	-	PF	-	CF
---	---	---	---	----	----	----	----	----	----	---	----	---	----	---	----

```
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0EA0 ES=0EA0 SS=0EA0 CS=0EA0 IP=0100 NV UP EI PL NZ NA PO NC
```

Gambar 3. Register Flags

Contoh instruksi MOV untuk transfer antar register dan pengisian langsung

```

-a
0F6C:0100 mov bx,abcd    → mengisi reg. BX dgn data ABCDH
0F6C:0103 mov ah,56      → mengisi reg. AH dgn data 56H
0F6C:0105 mov bl,ah       → mengcopy isi reg. AH ke reg. BL
0F6C:0107 mov ax,bx       → mengcopy isi reg. BX ke reg. AX
0F6C:0109
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0100 NV UP EI PL NZ NA PO NC
0F6C:0100 BCDAB          MOV     BX,ABCD
-t
AX=0000 BX=ABCD CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0103 NV UP EI PL NZ NA PO NC
0F6C:0103 B456          MOV     AH,56

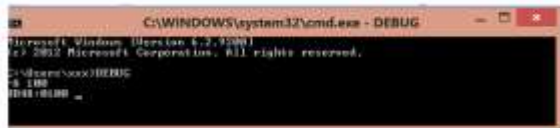
```

Gambar 4. Pengisian Data Register

HASIL DAN PEMBAHASAN

Untuk pengisian data Register Data pada register 32 bit dapat dilakukan dengan Penggabungan perintah register 8 bit dan register 16 bit, *software* yang digunakan adalah *Run, Cmd* dan bisa juga digunakan *Command Prompt*.

Langkah pertama dengan penginputan *Debug* dan perintah: a



Gambar 5. Memulai Program cmd

```

-t
AX=5600 BX=ABCD CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0105 NV UP EI PL NZ NA PO NC
0F6C:0105 8B56          MOV     BL,AH
-t
AX=5600 BX=AB56 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0107 NV UP EI PL NZ NA PO NC
0F6C:0107 8B0B          MOV     AX,BX
-t
AX=AB56 BX=AB56 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0F6C ES=0F6C SS=0F6C CS=0F6C IP=0109 NV UP EI PL NZ NA PO NC

```

```

C:\>cmd
Instruksi berikut akan menimbulkan Error :
mov ch,5678 : tidak bisa karena CH = 1 byte dan datanya 2 byte
mov dl,ax   : tidak bisa karena AX = 2 byte dan DL = 1 byte
mov dx,al   : tidak bisa karena AL = 1 byte dan DX = 2 byte

```

Gambar 6. Hasil Pengisian Register Alamat

Pengisian pada register Ax pada da 32 bit, dengan menginputkan register BH = untuk 8 bit, register BL untuk 16 bit dengan angka.Begitu seterusnya untuk pengisian register BX,CX,DX.

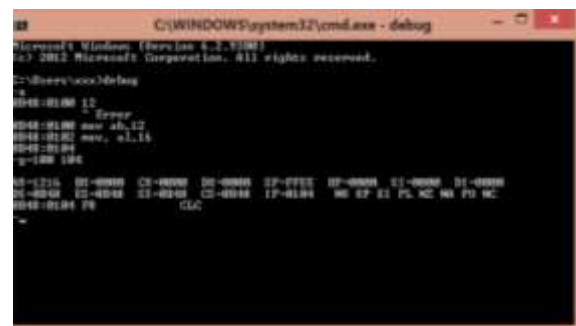
Untuk Pengisian Register Ax, dapat diinputkan dengan perintah :

Mov ah, 12

Mov al,16

g = 100 102

Hasilnya dapat dilihat pada gambar 7.



Gambar 7. Pengisian Register Data Ax

Kemudian seterusnya untuk penginputan register data BX, dapat dilakukan dengan penggabungan perintah yg digunakan pada register data untuk 32 bit.

55

DAFTAR PUSTAKA

Kurniawan, I. 2012. *Diktat Sistem Mikroprosesor*. Jambi: Politeknik Jambi.

Suprpto. 2008. *Bahasa Pemrograman Untuk SMK*, Jakarta: Direktorat Pembinaan Sekolah Menengah Kejuruan, Direktorat Jenderal

Manajemen Pendidikan Dasar dan Menengah Departemen Pendidikan Nasional.

Susanto 2008. *Sistem Mikroprosesor*, Cimahi:, Universitas Jenderal Achmad Yani Cimahi.

