Journal of Science and Social Research ISSN 2615 – 4307 (Print)
August 2018, I (2): 141 – 147 ISSN 2615 – 3262 (Online)
Available online at http://jurnal.goretanpena.com/index.php/JSSR

PENGISIAN DATA REGISTER DATA 8, 16, 32 BIT DAN OPERASI ARITMATIKA REGISTER MIKROPROSESOR

[1]Ikhsan Parinduri, [2]Siti Nurhabibah Hutagalung
[1]Sistem Komputer, STMIK Royal, Kisaran
[2]Teknik Informatika, STMIK Budidarma
e-mail: [1]Ikhsanparinduri9@gmail.com
[2]Siti_nurhabibah69@yahoo.com

Abstract: *Charging data registers on the microprocessor, can be done by combining between data registers 8, 16 and 32 bits. This can be done using cmd. Data registers for 32 bits, that is, a merger between 8bit data registers and 16 bit data registers. Register data AX (AH, AL), BX (BH, BL), CX (CH, CL). The proof can be done with several commands using assembly language, there are also arithmetic operations which include Addition (ADD), Pengurangaan (SUB), Multiplication (MUL) and Division (DIV).*

Keywords: Data Register Filling, Arithmetic Operations, Microprocessors

Abstrak: Pengisian register data pada mikroprosesor, dapat dilakukan dengan penggabungan antara register data 8, 16 dan 32 bit. Hal ini dapat dilakukukan dengan menggunakan *cmd*. Register data untuk 32 bit yaitu adalah penggabungan antara register data 8bit dengan register data 16 bit. Register data AX (AH,AL), BX (BH,BL), CX (CH,CL). Pembuktiannya dapat dilakukan dengan beberapa perintah menggunakan bahasa *assembly* juga terdapat operasi aritmatika yang diantaranya Penambahan (ADD), Pengurangaan (SUB), Perkalian (MUL) dan Pembagian (DIV).

Kata kunci: Pengisian Data Register, Operasi Aritmatika, Mikroprosesor

PENDAHULUAN

Mikroprosessor dapat dikelompokkan menurut teknologi yang dipergunakan, menurut jumlah bit data, menurut struktur atau menurut karakteristik mikroprosessor dan menurut fungsi dari mikroprosessor itu sendiri.

Berdasarkan jumlah bit data (*Word Size*) pada waktu ini telah terdapat banyak macam mirkroprosesor, mulai dari mikroprosessor 1bit, 4 bit, 8 bit,16 bit, 32 bit dan 64 bit.

Register merupakan sebagian memori dari mikroprosesor yang dapat diakses dengan kecepatan yang sangat tinggi.

Dalam melakukan pekerjaannya mikroprosesor selalu menggunakan register-register sebagai perantaranya, jadi register dapat diibaratkan sebagai aki dan tangannya mikroprosesor.

Pada bahasa *assembly* juga terdapat operasi matematika atau aritmatika yang diantaranya penambahan, pengurangaan, pembagiaan, perkalian.

Secara umum register-register dalam kelompok ini dapat digunakan untuk berbagai keperluan, walaupun demikian ada pula penggunaan khusus dari masingmasing register ini yaitu :

Journal of Science and Social Research ISSN 2615 – 4307 (Print)
August 2018, I (2): 141 – 147 ISSN 2615 – 3262 (Online)
Available online at http://jurnal.goretanpena.com/index.php/JSSR

Register AX, secara khusus digunakan pada operasi aritmatika terutama dalam operasi pembagian dan pengurangan.

Register BX, biasanya digunakan untuk menunjukkan suatu alamat *offset* dari suatu segmen.

Register CX, digunakan secara khusus pada operasi *looping* dimana register ini menentukan berapa banyaknya looping yang akan terjadi.

Register DX, digunakan untuk menampung sisa hasil pembagian 16 bit.

Pada prosesor 80386 terdapat tambahan register 32 bit, yaitu EAX, EBX, ECX dan EDX.

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

Perlu anda perhatikan bahwa pada perintah ADD ini antara Tujuan dan Asal harus mempunyai daya tampung yang sama, misalnya register AH (8 bit) dan AL (8 bit), AX (16 bit) dan BX (16 bit).

CONTOH (CODING):

; Menambah angka 74H pada ADD AL,74H ; isi AL. Hasil dalam AL

; Menambah jumlah BL ditambah membawa status ADD DX, BX ; pada isi DX

; Menambah huruf dari memory pada *offset* [SI] ADD DX,[SI] ; dalam DS ke isi DX

; Menambah *byte* dari PRICES [BX] ADD PRICES [BX],AL; isi *memory* pada *address* efektif ; PRICES [BX]

CONTOH (NUMERIK)

Penambahan angka-angka yang tidak ditandai; CL = 01110011 = 115 desimal + BL = 01001111 = 79 desimal ADD CL,BL ;

Hasil dalam CL ; CL = 11000010 = 194 desimal. Penambahan angka-angka yang ditandai CL = 01110011 = + 115 desimal + BL = 01001111 = + 79 desimal ADD CL,BL ; Hasil dalam CL; CL = 11000010 = - 62 desimal; salah sebab hasil terlalu besar untuk ukuran 7 bit.

ADC

Perintah ADC digunakan dengan cara yang sama pada perintah ADD, yaitu : ADC Tujuan,Asal.

Perbedaannya pada perintah ADC ini Tujuan tempat menampung hasil pertambahan Tujuan dan Asal ditambah lagi dengan *carry flag* (Tujuan: =Tujuan+Asal+Carry).

Pertambahan yang demikian bisa memecahkan masalah seperti yang pernah kita kemukakan, seperti pertambahan pada bilangan 12345678h+9ABCDEF0h.

Seperti yang telah kita ketahui bahwa satu register hanya mampu menampung 16 bit, maka untuk pertambahan seperti yang diatas bisa anda gunakan perintah ADC untuk memecahkannya,

Contoh:

MOV AX,1234h ; AX = 1234h CF = 0
MOV BX,9ABCh ; BX = 9ABCh CF = 0
MOV CX,5678h ; BX = 5678h CF = 0
MOV DX,0DEF0h ; DX = DEF0h CF = 0
ADD CX,DX ; CX = 3568h CF = 1
ADC AX,BX ; AX = AX+BX+CF = ACF1.

Hasil penjumlahan akan ditampung pada register AX:CX yaitu ACF13568h. Adapun *flag-flag* yang terpengaruh oleh perintah ADD dan ADC ini adalah CF,PF,AF,ZF,SF dan OF.

Journal of Science and Social Research ISSN 2615 – 4307 (Print)
August 2018, I (2): 141 – 147 ISSN 2615 – 3262 (Online)
Available online at http://jurnal.goretanpena.com/index.php/JSSR

.

OPERASI PENGURANGAN

CONTOH:
ASCII 9 – ASCII 5 (9 – 5)
AL = 00111001 = 39H = ASCII 9
BL = 00110101 = 35H = ASCII 5
SUB AL,BL ; Hasil: AL = 00000100 = BCD 04 dan CF = 0 AAS ; Hasil: AL = 00000100 = BCD 04 ; dan CF = 0; tidak diperlukan peminjaman ASCII 5 – ASCII 9 (5 – 9), Anggap AL = 00111001 = 35H = ASCII 5 ; dan BL = 00110101 = 39H = ASCII 9. SUB AL, BL ; Hasil: AL = 11111100 = - 4 ;

Dalam komplemen 2 dan CF = 1 AAS ; Hasil: AL = 00000100 = BCD 04 ; dan CF = 1; diperlukan peminjaman Instruksi AAS menyisakan hasil dari *hasil* BCD yang benar pada angka bawah AL dan menghapus angka atas AL pada semua 0.

Jika Anda ingin mengirim kembali ke sebuah terminal CRT, Anda bisa melakukan OR AL dengan 30H untuk menghasilkan kode ASCII yang benar sebagai hasilnya.

Jika perkalian digit angka dicari hasil pengurangannya, CF bisa diambil menjadi *account* dengan menggunakan instruksi SBB ketika mengurangi digit berikutnya.

Instruksi AAS hanya berfungsi untuk *register* AL. Instruksi ini memperbaharui AF dan CF tetapi OF, PF, SF, dan ZF tidak diterang 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 menolkan suatu register bisa anda kurangkan dengan dirinya sendiri seperti SUB AX,AX.

SBB

Seperti pada operasi penambahan, maka pada operasi pengurangan dengan bilangan yang besar(lebih dari 16 bit), bisa anda gunakan perintah SUB disertai dengan SBB (*Substract With Carry*). Perintah SBB digunakan dengan syntax:

SBB Tujuan,Asal

Perintah SBB akan mengurangkan nilai Tujuan dengan Asal dengan cara yang sama seperti perintah SUB, kemudian hasil yang didapat dikurangi lagi dengan *Carry Flag* (Tujuan:=Tujuan-Asal-CF).

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 varibel. 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 aka 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.

Sebelum Anda bisa mengalikan dua digit ASCII, Anda harus menutup 4

Journal of Science and Social Research ISSN 2615 – 4307 (Print)
August 2018, I (2): 141 – 147 ISSN 2615 – 3262 (Online)
Available online at http://jurnal.goretanpena.com/index.php/JSSR

bit angka atas dari masing-masing digit. Hal ini menyebabkan *hasil* BCD (satu digit BCD per *byte*) pada setiap *byte*.

Setelah dua *hasil* digit BCD dikalikan, instruksi AAM digunakan untuk mengatur hasil dari dua *hasil* digit BCD dalam AX.

AAM berfungsi hanya setelah pengalian dari dua *hasil byte* BCD, dan AAM hanya berfungsi hanya pada suatu *operand* dalam AL.

AAM memperbaharui PF, SF, dan ZF, tetapi AF, CF, dan OF tidak diterangkan.

CONTOH:

; AL = 00000101 = *hasil* BCD 5
; BH = 00001001 = *hasil* BCD 9
MUL BH ; AL x BH ; hasil dalam AX
; AX = 00000000 00101101 = 002DH
AAM ; AX = 00000100 00000101 = 0405H, yang merupakan *hasil* BCD untuk angka 45.

Jika menginginkan kode ASCII untuk hasilnya, gunakan instruksi berikutnya OR AX, 3030H ; Letakkan 3 pada angka atas di setiap *byte.*
; AX = 00110100 00110101 = 3435H,
; yang merupakan kode ASCII untuk angka 45.

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.

Bila **sumber** merupakan operand 16 bit seperti **DIV BX**, maka komputer akan mengambil nilai yang terdapat pada register DX:AX dan membaginya dengan nilai BX. Hasil pembagian 16 bit ini akan disimpan pada register AX dan sisa dari pembagian akan disimpan pada register DX.

AAD mengubah dua *hasil* digit BCD menjadi AH dan AL menjadi angka biner yang seimbang dalam AL. Pengaturan ini harus dibuat sebelum membagi dua *hasil* digit BCD dalam AX dengan *byte hasil* BCD.

Setelah pembagian, AL akan berisi hasil bagi dari *hasil* BCD dan AH akan berisi sisa *hasil* BCD. PF, SF, dan ZF diperbaharui.
AF, CF, dan OF tidak diterangkan setelah AAD.

CONTOH:

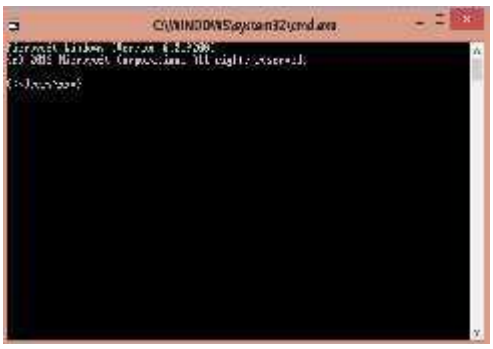
; AX = 0607H *hasil* BCD untuk 67 desimal
; CH = 09H, sekarang atur menjadi biner
AAD ; Hasil: AX = 0043 = 43H = 67 desimal
DIV CH ; Bagi AX dengan *hasil* BCD pada CH
Hasil Bagi : AL = 07 *hasil* BCD
Sisa : AH = 04 *hasil* BCD
Flags tidak diterangkan setelah DIV.

METODE

Penelitian ini bertujuan untuk pembuktian operasi aritmatika bahasa *assembly* pada register mikroprosesor.

Dalam hal ini pembuktian menggunakan program yang terdapat pada komputer yaitu *Cmd*.
Dengan perintah sebagai berikut :

1. Masukkan perintah pada *Run* **: *cmd***
   Akan tampil pada layar :



Gambar 1. Tampilan Awal *Cmd*

Journal of Science and Social Research ISSN 2615 – 4307 (Print)
August 2018, I (2): 141 – 147 ISSN 2615 – 3262 (Online)
Available online at http://jurnal.goretanpena.com/index.php/JSSR

Cara menggunakan :

1. Di lingkungan Windows, klik Start‟ dan kemudian pilih Run. Dari window Run ketikkan Debug dan klik tombol OK.
2. Di lingkungan DOS, pindahlah ke *subdirectory* yang berisi instruksi-instruksi DOS.
   Jika Windows terinstall, pindahlah ke C:\Windows\Command, dan kemudian ketik Debug dan tekan tombol Enter.

Salah satu dari kedua cara tersebut akan memanggil program Debug dengan menampilkan *cursor* berbentuk *strip* (-)

2. Kemudian masukkan perintah debug dan perintah a100 :
3. 

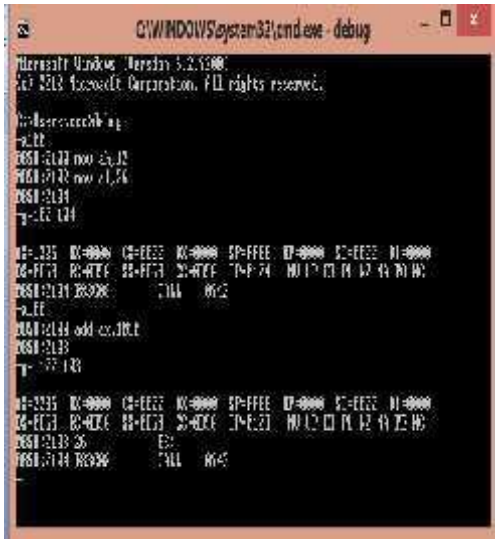
Gambar 2. Penginputan Nilai Untuk Pengisian Register

4. Lakukan pengisian register dengan perintah MOV pada register AX, BX, CX, DX :
   a. Untuk Ax, dengan mengetikkan perintah : MOV AH, 10
      MOV AL,
      12 Enter
      g= 100 104

Terlihat pada tampilan berikut ini :

Gambar 3. Pengisian Register AX

5. Kemudian Lakukan Pengisian untuk register BX, CX, DX
   Terlihat pada tampilan berikut ini :



Gambar 4. Pengisian Register BX,CX, DX

Journal of Science and Social Research ISSN 2615 – 4307 (Print)
August 2018, I (2): 97 – 102 ISSN 2615 – 3262 (Online)
Available online at http://jurnal.goretanpena.com/index.php/JSSR

HASIL DAN PEMBAHASAN

Untuk Operasi Aritmatika pada Register dapat dilakukan dengan beberapa perintah diantaranya adalah :

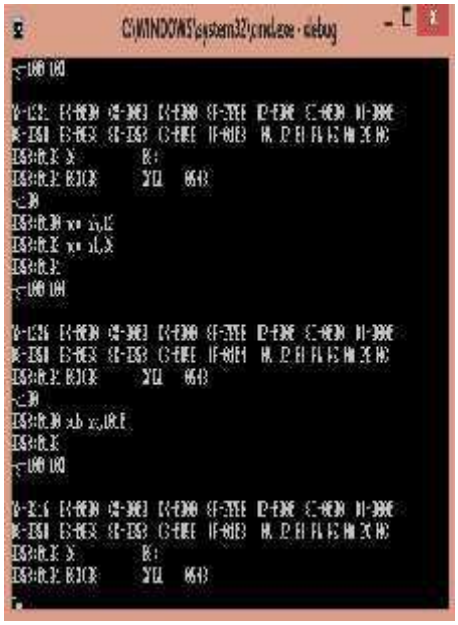
1. **Penjumlahan Register**,
   untuk Register AX, dengan Ah, Al, dengan perintah menggunakan ADD AX.
   Perintah yang dilakukan dengan mengetikkan :
   Debug
   Lakukan pengisian data register AX,
   -a100
   MOV AH,12
   MOV AL, 26
   g=100 105
   Lakukan dengan Operasi Penjumlahan dengan perintah :

ADD AX, 1010
Enter
g= 100 103
Artinya adalah Register AX =1226 dijumlahkan dengan bilangan 1010 adalah
Terlihat pada gambar berikut :

Gambar Penjumlahan pada Register BX,CX, DX

2. Pengurangan Register, untuk Register AX, dengan AH, AL, dengan perintah menggunakan SUB AX.
   Perintah yang dilakukan dengan mengetikkan :
   Debug
   Lakukan pengisian data register AX,
   -a100
   MOV AH,12
   MOV AL, 26
   g=100 104

Lakukan dengan Operasi Pengurangan dengan perintah :
SUBB AX, 1010
Enter
g= 100 103
Artinya adalah Register AX =1226
Terlihat pada gambar berikut :



Gambar 5. Pengurangan pada Register BX,CX, DX

3. Perkalian Register, untuk Register AX, dengan AH, AL, dengan perintah menggunakan SUB AX.
   Perintah yang dilakukan dengan mengetikkan :
   Debug
   Lakukan pengisian data register AX,
   -a100
   MOV AH,12
   MOV AL, 26
   g=100 104

Lakukan dengan Operasi Perkalian dengan perintah :
MUL AX, 1010
Enter
g= 100 102
Terlihat pada gambar berikut :

Journal of Science and Social Research ISSN 2615 – 4307 (Print)
August 2018, I (2): 97 – 102 ISSN 2615 – 3262 (Online)
Available online at http://jurnal.goretanpena.com/index.php/JSSR

Gambar 6. Perkalian pada Register BX,CX, DX

SIMPULAN

Hasil dari penelitian pengisian data register data pada mikroprosesor dan kemudian dilakukannya operasi aritmatika dapat ditarik beberapa kesimpulan diantaranya adalah:

1. Pengisian data Register menggunakan *Cmd*, bertujuan untuk mengetahui tentang penggabungan pengisian data pada register 8 bit (AH) dan 16 bit (AL) menjadi 32 bit (AX)
2. Pembuktian operasi aritmatika (Penjumlahan, Pengurangan, Perkalian dan Pembagian) pada register.

DAFTAR PUSTKA

Douglas V. Hall, "*Microprocessors and Interfacing : Programming and Hardware*", 2nd ed, McGraw Hill

Intel, "8088 *Data Sheet Book*", Agustus 1990

Irwan Kurniawan, 2012, Diktat Sistem Mikroprosesor, Politeknik Jambi.

Putu Sudira, 2015 Sistem Mikroprosesor dan Mikrokontroler. Diknik Elektronika FT UNY.

Sanjiva Nath, "*Assembly Language Interfacing in Turbo Pascal*", MIS Press, 1987

Susanto Sambasri, Rizal Suryana, 2008. Diktat Kuliah Sistem Mikroprosesor, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jenderal Achmad Yani, Cimahi.