

LAPORAN PRAKTIKUM
BAHASA RAKITAN



Disusun Oleh :

Nama : Indah Cahya Resti

NIM : 09011281823046

Jurusan : Sistem Komputer

Dosen : Aditya P. P. Prasetyo, S. Kom.,M.T.

LABORATORIUM ROBOTIKA DAN SISTEM KENDALI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA
INDRALAYA 2020

PRAKTIKUM II

OPERASI ARITMATIKA

A. TUJUAN

- Mengerti dan memahami prinsip dasar dari proses operasi aritmatika
- Mengerti dan memahami setiap instruksi yang digunakan dalam proses operasi aritmatika
- Mampu menganalisa setiap data yang dihasilkan

B. PERALATAN YANG DIGUNAKAN

- Seperangkat PC
- Software *Command Prompt*
- Modul *Praktikum Bahasa Rakitan*

C. PROSEDURE PRAKTIKUM

- Salinlah program 2
- Amatilah output yang ditampilkan
- Buat Algoritma program
- Buat Flowchart program
- Buat Analisis program
- Buat Kesimpulan

D. DASAR TEORI

Dalam operasi aritmatika terdapat beberapa instruksi, seperti berikut:

- **Penjumlahan (ADD)**

Bilangan 8 bit atau isi dari suatu register atau isi dari suatu memori dapat dikurangkan dengan Accumulator dan hasilnya disimpan dalam accumulator.

MOV 35H,#20H : Isi lokasi memori 35H dengan data 20H

ADD A,35H : Kurangkan data 20H dengan data yang ada didalam Accumulator

- **Pengurangan (SUBB)**

Bilangan 8 bit atau isi dari suatu register atau isi dari suatu memori dapat dikurangkan dengan Accumulator dan hasilnya disimpan dalam accumulator.

MOV Rn,#21H : Isi data 21H ke register Rn

SUBB A,Rn : Kurangkan data accumulator dengan data yang ada di register Rn

- **Penaikan Data (INC)**

Bilangan 8 bit atau isi dari suatu register atau isi dari suatu memori dapat dinaikkan 1 bit.

INC A : naikkan isi accumulator

INC 80H : naikkan isi memori 80H

INC Rn : naikkan isi register Rn

- **Penurunan Data (DEC)**

Bilangan 8 bit atau isi dari suatu register atau isi dari suatu memori dapat diturunkan 1 bit.

DEC A : turunkan 1 bit isi accumulator

DEC 80H : turunkan 1 bit isi memori 80H

DEC Rn : turunkan 1 bit isi register Rn

Flag-flag register adalah :

- OF (*Over Flow Flag*), jika terjadi over flow pada operasi aritmatika, bit ini akan bernilai 1
- SF (*Sign Flag*), jika digunakan bilangan bertanda, bit ini akan bernilai 1
- ZF (*Zerro Flag*), jika hasil operasi menghasilkan nol, bit ini akan bernilai 1
- CF (*Carry Flag*), jika terjadi borrow pada operasi pengurangan atau carry pada penjumlahan, bit ini akan bernilai 1
- PF (*Parity Flag*), digunakan untuk menunjukkan paritas bilangan. Bit ini akan bernilai 1 jika bilangan yang dihasilkan merupakan bilangan genap dan akan bernilai 0 jika bilangan yang dihasilkan merupakan bilangan ganjil.
- DF (*Direction Flag*), digunakan pada operasi string untuk menunjukkan arah proses.
- IF (*Interrupt Enable Flag*), CPU akan mengabaikan interupsi yang terjadi jika flag ini bernilai 0
- TF (*Trap Flag*), digunakan untuk debugging, dengan operasi step by step
- AF (*Auxiliary Flag*), digunakan pada operasi BCD

E. PROGRAM

Program 2

```
MOV AL,11
MOV BL,22
MOV CL,33
MOV DL,44
INC CH
INC DH
ADD AL,BL
ADD DL,23
DEC DH
ADD BL,32
ADD AX,BX
SUB AL,BL
SUB BX,0808
DEC CX
SBB DL,23
```

F. OUTPUT PROGRAM

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
-T
AX=0011 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=072A ES=072A SS=072A CS=072A IP=0102 NU UP EI NG NZ NA PO NC
072A:0102 B322          MOV     BL,22
-T
AX=0011 BX=0022 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=072A ES=072A SS=072A CS=072A IP=0104 NU UP EI NG NZ NA PO NC
072A:0104 B133          MOV     CL,33
-T
AX=0011 BX=0022 CX=0033 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=072A ES=072A SS=072A CS=072A IP=0106 NU UP EI NG NZ NA PO NC
072A:0106 B244          MOV     DL,44
-T
AX=0011 BX=0022 CX=0033 DX=0044 SP=FFFE BP=0000 SI=0000 DI=0000
DS=072A ES=072A SS=072A CS=072A IP=0108 NU UP EI NG NZ NA PO NC
072A:0108 FEC5          INC     CH
-T
AX=0011 BX=0022 CX=0133 DX=0044 SP=FFFE BP=0000 SI=0000 DI=0000
DS=072A ES=072A SS=072A CS=072A IP=010A NU UP EI PL NZ NA PO NC
072A:010A FEC6          INC     DH
-T
AX=0011 BX=0022 CX=0133 DX=0144 SP=FFFE BP=0000 SI=0000 DI=0000
DS=072A ES=072A SS=072A CS=072A IP=010C NU UP EI PL NZ NA PO NC
072A:010C 00DB          ADD     AL,BL
-
```

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
-T
AX=0011 BX=0022 CX=0133 DX=0144 SP=FFFE BP=0000 SI=0000 DI=0000
DS=072A ES=072A SS=072A CS=072A IP=010C NU UP EI PL NZ NA PO NC
072A:010C 00DB          ADD     AL,BL
-T
AX=0033 BX=0022 CX=0133 DX=0144 SP=FFFE BP=0000 SI=0000 DI=0000
DS=072A ES=072A SS=072A CS=072A IP=010E NU UP EI PL NZ NA PE NC
072A:010E B0C223        ADD     DL,23
-T
AX=0033 BX=0022 CX=0133 DX=0167 SP=FFFE BP=0000 SI=0000 DI=0000
DS=072A ES=072A SS=072A CS=072A IP=0111 NU UP EI PL NZ NA PO NC
072A:0111 FECE          DEC     DH
-T
AX=0033 BX=0022 CX=0133 DX=0067 SP=FFFE BP=0000 SI=0000 DI=0000
DS=072A ES=072A SS=072A CS=072A IP=0113 NU UP EI PL ZR NA PE NC
072A:0113 B0C332        ADD     BL,32
-T
AX=0033 BX=0054 CX=0133 DX=0067 SP=FFFE BP=0000 SI=0000 DI=0000
DS=072A ES=072A SS=072A CS=072A IP=0116 NU UP EI PL NZ NA PO NC
072A:0116 01DB          ADD     AX,BX
-T
AX=0087 BX=0054 CX=0133 DX=0067 SP=FFFE BP=0000 SI=0000 DI=0000
DS=072A ES=072A SS=072A CS=072A IP=0118 NU UP EI PL NZ NA PE NC
072A:0118 2BD8          SUB     AL,BL
-
```

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
-T
AX=0033 BX=0054 CX=0133 DX=0067 SP=FFFE BP=0000 SI=0000 DI=0000
DS=072A ES=072A SS=072A CS=072A IP=0116 NU UP EI PL NZ NA PO NC
072A:0116 01D8          ADD     AX,BX
-T
AX=0087 BX=0054 CX=0133 DX=0067 SP=FFFE BP=0000 SI=0000 DI=0000
DS=072A ES=072A SS=072A CS=072A IP=0118 NU UP EI PL NZ NA PE NC
072A:0118 28D8          SUB     AL,BL
-T
AX=0033 BX=0054 CX=0133 DX=0067 SP=FFFE BP=0000 SI=0000 DI=0000
DS=072A ES=072A SS=072A CS=072A IP=011A NU UP EI PL NZ NA PE NC
072A:011A 81EB0808      SUB     BX,0808
-T
AX=0033 BX=F84C CX=0133 DX=0067 SP=FFFE BP=0000 SI=0000 DI=0000
DS=072A ES=072A SS=072A CS=072A IP=011E NU UP EI NG NZ AC PO CY
072A:011E 49           DEC     CX
-T
AX=0033 BX=F84C CX=0132 DX=0067 SP=FFFE BP=0000 SI=0000 DI=0000
DS=072A ES=072A SS=072A CS=072A IP=011F NU UP EI PL NZ NA PO CY
072A:011F 80DA23      SBB     DL,23
-T
AX=0033 BX=F84C CX=0132 DX=0043 SP=FFFE BP=0000 SI=0000 DI=0000
DS=072A ES=072A SS=072A CS=072A IP=0122 NU UP EI PL NZ NA PO NC
072A:0122 0000          ADD     [BX+SI],AL
DS:F84C=00

```

TAMPILKAN OUTPUT SCREEN

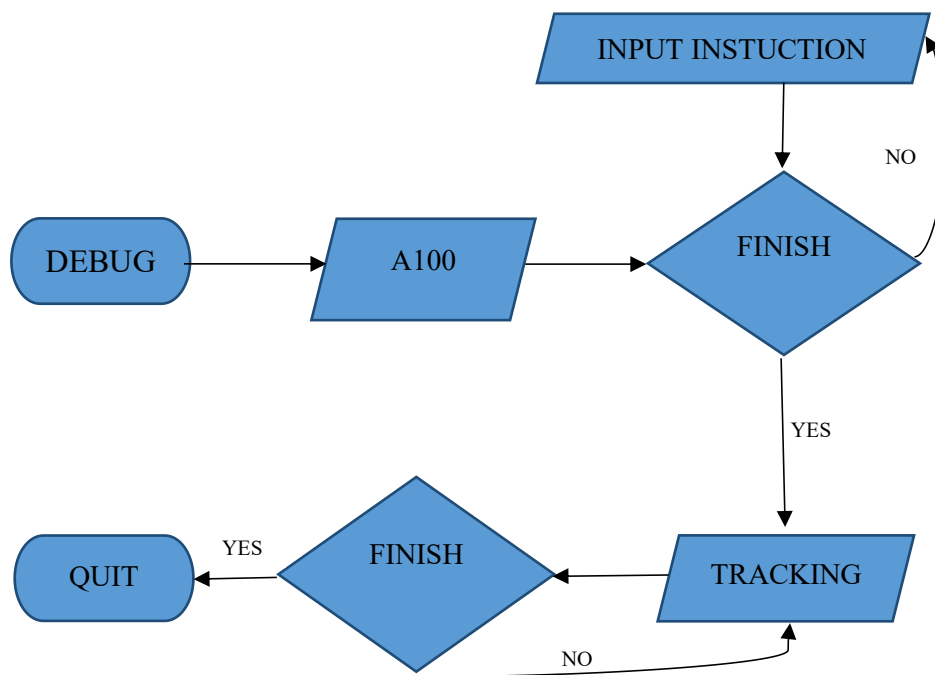
Tabel Data

Alamat	Code	Mnemonic (Operand)	AX		BX		CX		DX	
			AH	AL	BH	BL	CH	CL	DH	DL
0100		MOV AL,11								
0102	B322	MOV BL,22	00	11	00	00	00	00	00	00
0104	B133	MOV CL,33	00	11	00	22	00	00	00	00
0106	B244	MOV DL,44	00	11	00	22	00	33	00	00
0108	FEC5	INC CH	00	11	00	22	00	33	00	44
010A	FEC6	INC DH	00	11	00	22	01	33	00	44
010C	00D8	ADD AL,BL	00	11	00	22	01	33	01	44
010E	80C223	ADD DL,23	00	33	00	22	01	33	01	44
0111	FECE	DEC DH	00	33	00	22	01	33	01	67
0113	80C332	ADD BL,32	00	33	00	22	01	33	00	67
0116	01D8	ADD AX,BX	00	33	00	54	01	33	00	67
0118	28D8	SUB AL,BL	00	87	00	54	01	33	00	67
011A	81EB0808	SUB BX,0808	00	33	00	54	01	33	00	67
011E	49	DEC CX	00	33	F8	4C	01	33	00	67
011F	80DA23	SBB DL,23	00	33	F8	4C	01	32	00	67
0122	0000		00	33	F8	4C	01	32	00	43

G. ALGORITMA PROGRAM

1. Lakukan DEBUG untuk masuk ke sebuah program yang akan dibuat.
2. Masukkan inputan A100 untuk memulai program di alamat offset 0100.
3. Periksa apakah input A100 telah dimasukkan. Jika sudah lanjutkan ke proses Tracking, apabila tidak diinput, lakukan instruksi inputan kembali.
4. Lakukan Tracking.
5. Periksa apakah input sebelumnya telah ditracking. Jika tidak, silahkan ulangi tracking tersebut. Apabila telah selesai tracking, silahkan “QUIT” untuk keluar dari program.

H. FLOWCHART PROGRAM



I. ANALISIS PRAKTIKUM

Pada praktikum ini menggunakan aplikasi DOSbox untuk membuat program dengan perintah MOV, ADD, INC, SUBB, dan DEC. Dimulai dengan DEBUG kemudian membuat program awal dengan menggunakan perintah A100 (A100 untuk memulai program yang terletak pada alamat offset 0100).



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
072A:0104 MOV CL,33
072A:0106 DL,44
^ Error
072A:0106
-QUIT
E:\>DEBUG
-A100
072A:0100 MOV AL,11
072A:0102 MOV BL,22
072A:0104 MOV CL,33
072A:0106 MOV DL,44
072A:0108 INC CH
072A:010A INC DH
072A:010C ADD AL,BL
072A:010E ADD DL,23
072A:0111 DEC DH
072A:0113 ADD BL,32
072A:0116 ADD AX,BX
072A:0118 SUB AL,BL
072A:011A SUB BX,0000
072A:011E DEC CX
072A:011F SBB DL,23
072A:0122
-
```

Di dalam program ini menggunakan beberapa operasi aritmatika. Sehingga pada table data dapat kita lihat perubahan yang dilakukan tiap perintah :

- Perintah MOV AL,11 melakukan pengisian data 11 ke register AL, sehingga selanjutnya pada alamat 0102 telah terisi data 11 di register AL.
- Perintah INC CH artinya menaikkan data 1 bit pada register CH, sehingga selanjutnya pada alamat 010A memiliki data 01 di register CH.
- Perintah ADD AL,BL artinya melakukan penambahan data dari register BL(22) ke register AL(11), sehingga selanjutnya pada alamat 010E telah terisi menjadi data 33 di register AL.
- Perintah DEC DH artinya menurunkan data 1 bit pada register DH, sehingga selanjutnya pada alamat 0113 memiliki data 00 di register DH.
- Perintah SUB AL,BL artinya melakukan pengurangan data dari register BL(54) ke register AL(87), sehingga selanjutnya pada alamat 011A telah terisi menjadi data 33 di register AL.
- Perintah SBB DL,23 artinya melakukan pengurangan data 23 ke register DL(67), sehingga selanjutnya pada alamat 0122 telah terisi menjadi data 43 di register DL.

J. KESIMPULAN

Jadi, hasil dari praktikum ini adalah melakukan pengolahan data menggunakan program DEBUG. Di dalam program ini menggunakan general purpose register, yaitu register AX (AH dan AL), BX (BH dan BL), CX (CH dan CL), dan DX (DH dan DL) dimana pada setiap register ini terdiri dari 16 bit yang memiliki fungsinya masing-masing.

Program ini menggunakan beberapa perintah, yaitu perintah MOV untuk memindahkan data register atau register ke register (contohnya memindahkan data 11 ke register AL), perintah INC untuk menaikkan data 1 bit pada register, perintah ADD untuk menambahkan/menjumlahkan data pada register ataupun menjumlahkan data dari register ke register yang lain, perintah DEC untuk menurunkan data 1 bit pada register, perintah SUB untuk mengurangi data pada register ataupun mengurangi data dari register ke register yang lain, dan perintah SBB untuk melakukan pengurangan data pada register.