

M. Faqih
2020230032

Praktikum Bahasa Rakitan

Jumat
15 - 04 - 2022

Praktikum II Penggunaan Lompatan Bersyarat dan Tidak
Bersyarat, Operasi Aritmatika dan Stack

Praktikum II

Penggunaan Lompatan Bersyarat dan Tidak Bersyarat, Operasi Aritmatika dan Stack

Tujuan : Agar mahasiswa mampu membuat program, membedakan lompatan bersyarat dan tidak bersyarat, konsep dasar operasi stack dan penggunaan operasi matematika untuk manipulasi bilangan.

Materi :

- Perbedaan lompatan bersyarat (Conditional Jump) dan lompatan tidak bersyarat (Unconditional Jump).
- Cara kerja stack melalui perintah PUSH dan POP.
- Operasi matematika dan pengaruhnya terhadap register flag dan register lain yang terlibat didalamnya.

Alat :

- IBM-PC.
- DOS versi 6.X keatas dengan file debug.exe.

A. Penggunaan lompatan bersyarat dan tidak bersyarat serta perintah-perintah yang mengikutinya.

1. Jalankan program debug.exe, melalui DOS-prompt C:\>debug <tekan enter>.
2. setelah tampil tanda hubung (-), ketikkan A100 <dan tekan enter>.
3. Tulislah listing program berikut ini :

CS:100 MOV CX,3
 MOV DL,41
 MOV AH,2

ULANG : INT 21

INC DL

LOKASI : LOOP ULANG
 INT 20

4. Jalankan program secara single step dengan perintah T=100 untuk eksekusi baris program berikutnya cukup tuliskan T saja dan tekan enter. Perhatikan perubahan isi register DL dan CX, dan lengkapi tabel data pengamatan berikut ini :

Tabel 2-1

Lokasi alamat	Source Code	AX		BX		CX		DX		CS	DS	IP	SI	DI
		AH	AL	BH	BL	CH	CL	DH	DL					
CS:100	MOV CX,3	00	00	00	00	00	03	00	00	073F	073F	0103	0000	0000
	MOV DL,41	00	00	00	00	00	03	00	41	073F	073F	0105	0000	0000
	MOV AH,2	02	00	00	00	00	03	00	41	073F	073F	0107	0000	0000
	INT 21	02	41	00	00	00	03	00	41	073F	073F	0109	0000	0000
	INC DL	02	41	00	00	00	03	00	42	073F	073F	010B	0000	0000
	LOOP Alamat	02	41	00	00	00	02	00	42	073F	073F	0107	0000	0000
	INT 20	02	43	00	00	00	00	00	44	F000	073F	010D	0000	0000

5. Eksekusi program dengan perintah G=100 <dan tekan enter>, tuliskan tampilan yang terdapat di layar monitor :

ABC

6. Gantilah listing program pada prosedur kerja no. 3 dengan program dibawah ini :

CS:100 MOV DL,41 → memindahkan hexa A ke DL
 MOV AH,2 → perintah mencetak pada layar
 LONCAT: INT 21 → eksekusi service yang ditentukan
 INC DL → operasi aritmatika, menambah nilai 1 DL. DL bertambah sampai memenuhi kondisi CMP terpenuhi
 CMP DL,44 → membandingkan DL jika terpenuhi program selesai
 JNE LONCAT → Selama nilai DL tidak 44, maka lontcat INT 21
 INT 20 → tanda akhir program

7. Eksekusi program diatas dengan perintah G=CS:100 <tekan enter>, maka

tampilan di layar monitor : ABC

8. Berikan penjelasan cara kerja setiap baris program.

9. Dari percobaan diatas, tuliskan mnemonic untuk conditional jump dan unconditional jump.

JMP

JNB, JNC

B. Cara kerja stack segment melalui perintah PUSH dan POP.

Stack segment digunakan untuk mencatat alamat segmen memori yang dipergunakan menjadi stack sebagai tempat simpan baca operasi PUSH dan POP.

1. Jalankan program debug.exe dari DOS-prompt, ketikkan C:\>Debug <tekan enter> setelah tampil tanda hubung (-), maka ketikkan A100 <tekan enter>, selanjutnya akan ditampilkan alamat awal CS:0100.
2. Tuliskan Source Code program assembly berikut ini :

```

CS:0100    MOV BP,200
           MOV SP,300
           MOV AL, [BP]
           MOV AH, [BP+1]
           MOV DL, [BP+2]
           MOV DH, [BP+3]
           MOV BL, [BP+4]
           MOV BH, [BP+5]
           PUSH BX
           PUSH AX
           PUSH DX
           MOV AX,5566
           MOV DX,7788
           POP SI
           POP DX
           POP AX
           INT 20

```

3. Isikan data dimulai dari alamat offset 200 dengan mengetikkan e200 11, 22, 33, 44, AA, BB <tekan enter>.
4. Jalankan program secara single step menggunakan perintah T=100 <tekan enter>, untuk eksekusi program pada baris berikutnya cukup tekan T saja.
5. Setelah perintah PUSH BX, dieksekusi maka isi register BH berada di memori yang alamatnya SS:02FFH dan isi register BL berada di memori yang alamatnya SS:02FEH. Untuk perintah PUSH AX ketika dieksekusi maka isi register AH berada di memori yang alamatnya SS:02FDH dan isi register AL berada di memori yang alamatnya SS:02FCH. Tentukanlah memori alamat untuk perintah PUSH DX, dan catatlah isi register BX, AX dan DX dengan perintah DSS:02F0H.

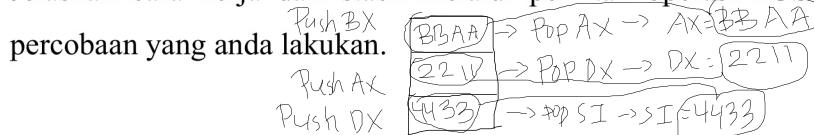
PUSH BX	SS:02FFH	BH → <u>B</u> <u>B</u>
	SS:02FEH	BL → <u>A</u> <u>A</u>
PUSH AX	SS:02FDH	AH → <u>2</u> <u>2</u>
	SS:02FCH	AL → <u>1</u> <u>1</u>
PUSH DX	SS:02FBH	DH → <u>4</u> <u>4</u>
	SS:02FAH	DL → <u>3</u> <u>3</u>

Lengkapi data pengamatan pada tabel 2-2 berikut ini :

Tabel 2-2

Lokasi alamat	Source Code	AX		BX		CX		DX		CS	DS	IP	SI	DI
		AH	AL	BH	BL	CH	CL	DH	DL					
CS:100	MOV BP,200	00	00	00	00	00	00	00	00	073F	073F	0103	0000	0000
	MOV SP,300	00	00	00	00	00	00	00	00	073F	073F	0106	0000	0000
	MOV AL,[BP]	00	11	00	00	00	00	00	00	073F	073F	0109	0000	0000
	MOV AH,[BP+1]	22	11	00	00	00	00	00	00	073F	073F	010C	0000	0000
	MOV DL,[BP+2]	22	11	00	00	00	00	00	33	073F	073F	010F	0000	0000
	MOV DH,[BP+3]	22	11	00	00	00	00	44	33	073F	073F	0112	0000	0000
	MOV BL,[BP+4]	22	11	00	AA	00	00	44	33	073F	073F	0115	0000	0000
	MOV BH,[BP+5]	22	11	BB	AA	00	00	44	33	073F	073F	0118	0000	0000
	PUSH BX	22	11	BB	AA	00	00	44	33	073F	073F	0119	0000	0000
	PUSH AX	22	11	BB	AA	00	00	44	33	073F	073F	011A	0000	0000
	PUSH DX	22	11	BB	AA	00	00	44	33	073F	073F	011B	0000	0000
	MOV AX, 5566	55	66	BB	AA	00	00	44	33	073F	073F	011E	0000	0000
	MOV DX, 7788	55	66	BB	AA	00	00	77	88	073F	073F	0121	0000	0000
	POP SI	55	66	BB	AA	00	00	77	88	073F	073F	0122	4433	0000
	POP DX	55	66	BB	AA	00	00	77	11	073F	073F	0123	4433	0000
	POP AX	BB	AA	BB	AA	00	00	22	11	073F	073F	0124	4433	0000
	INT 20	BB	AA	BB	AA	00	00	22	11	073F	073F	1480	4433	0000

6. Jelaskan cara kerja dari stack melalui perintah operasi PUSH dan POP dari



C. Memahami operasi matematika pada bahasa assembly dan pengaruhnya terhadap flag register.

- Jalankan program debug.exe dari DOS-prompt, ketikkan C:\>debug <tekan enter>, setelah tampil tanda hubung (-), ketikkan A100 <tekan enter> dan akan muncul alamat offset awal CS:0100.
- Tuliskan source Code bahasa assembly berikut ini :

CS:0100 CLC	MOV BL, 10
MOV AL, 3D	MUL BL
MOV DL,75	MOV AL,07
ADD AL,DL	NEG AL
MOV BX,140	MOV AL, 0010110B
MOV AL, [BX] NOT AL	

```

ADC AL,[BX+1] MOV DI,0011H
MOV [BX+3],AL OR DI, 1101H
MOV SI, 200      INT 20
SUB SI, 350
MOV AL,20

```

3. Isikan data untuk alamat offset 140, dengan perintah e140 92,7F <tekan enter>.

4. Lengkapi tabel 2-3 dari data pengamatan berikut ini :

Tabel 2-3

Lokasi alamat	Source Code	AX		BX		CX		DX		CS		DS		IP	SI	DI
		AH	AL	BH	BL	CH	CL	DH	DL							
CS:100	CLC	00	00	00	00	00	00	00	00	073F	073F	0101	0000	0000		
	MOV AL,3D	00	BD	00	00	00	00	00	00	073F	073F	0103	0000	0000		
	MOV DL,75	00	3D	00	00	00	00	00	75	073F	073F	0105	0005	0000		
	ADD AL,DL	00	B2	00	00	00	00	00	75	073F	073F	0107	0000	0000		
	MOV BX,140	00	B2	01	40	00	00	00	75	073F	073F	010A	0000	0000		
	MOV AL,[BX]	00	92	01	40	00	00	00	75	073F	073F	010C	0000	0000		
	ADC AL,[BX+1]	00	11	01	40	00	00	00	75	073F	073F	010F	0000	0000		
	MOV [BX+3],AL	00	11	01	40	00	00	00	75	073F	073F	0112	0000	0000		
	MOV SI,200	00	11	D1	40	10	00	00	75	073F	073F	0115	0200	0000		
	SUB SI,350	00	11	01	40	00	00	00	75	073F	073F	0119	FEB0	0000		
	MOV AL,20	00	2B	01	40	00	00	00	75	073F	073F	011B	FEB0	0000		
	MOV BL,10	00	20	01	10	00	00	00	75	073F	073F	011D	FEB0	0000		
	MUL BL	02	00	01	10	00	00	00	75	073F	073F	011F	FEB0	0000		
	MOV AL,07	02	07	01	10	00	00	00	75	073F	073F	0121	FEB0	0000		
	NEG AL	02	F9	01	10	00	00	00	75	073F	073F	0123	FEB0	0000		
	MOV AL,000010110B	02	16	01	10	00	00	00	75	073F	073F	0125	FEB0	0000		
	NOT AL	02	E9	01	10	00	00	00	75	073F	073F	0127	FEB0	0000		
	MOV DI,0011H	02	E9	01	10	00	00	00	75	073F	073F	012A	FEB0	0011		
	OR DI,1101H	02	E9	01	10	00	00	00	75	073F	073F	012E	FEB0	1111		
	INT 20	02	E9	01	10	00	00	00	75	073F	073F	1480	FEB0	1111		

Tabel 2-4 Penjelasan nilai setiap bit flag

Register Flag																	
OF	DF	IF		TF		SF		ZF		AF		PF		CF			
OV	NV	DN	UP	EI	DI			NG	PL	ZR	NZ	AC	NA	PE	PO	CY	NC
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Keterangan : 1 = SET, 0 = RESET																	

Tabel 2-5 Data Pengamatan untuk register flag

Lokasi Alamat	Source Code	Register Flag									
		OF	DF	IF	TF	SF	ZF	AF	PF	CF	
CS:0100	CLC	0	0	1	1	0	0	0	0	0	
	MOV AL,3D	0	0	1	1	0	0	0	0	0	
	MOV DL,75	0	0	1	1	0	0	0	0	0	
	ADD AL,DL	1	0	1	1	0	1	1	0	0	
	MOV BX,140	1	0	1	1	0	1	1	0	0	
	MOV AL,[BX]	1	0	1	1	0	1	1	0	0	
	ADC AL,[BX+1]	0	0	1	1	0	0	1	1	1	
	MOV [BX+3],AL	0	0	1	1	0	0	1	1	1	
	MOV SI,200	0	0	1	1	0	0	1	1	1	
	SUB SI,350	0	0	1	1	0	0	0	1	1	
	MOV AL,20	0	0	1	1	0	0	0	1	1	
	MOV BL,10	0	0	1	1	0	0	0	1	1	
	MUL BL	1	0	1	1	0	1	0	1	1	
	MOV AL,07	1	0	1	1	0	1	0	1	1	
	NEG AL	0	0	1	1	0	1	1	1	1	
	MOV AL,00010110B	0	0	1	1	0	1	1	1	1	
	NOT AL	0	0	1	1	0	1	1	1	1	
	MOV DI,0011H	0	0	1	1	0	1	1	1	1	
	OR DI,1101H	0	0	1	1	0	0	1	0	0	
	INT 20	0	0	0	0	0	0	0	1	0	

5. Laksanakan tugas keterampilan untuk membuat program aplikasi penggunaan perintah PUSH, POP dan operasi matematika, bertanyalah kepada asisten untuk bentuk tugasnya.