

**LAB SHEET 5**  
**PEMROGRAMAN DENGAN STACK POINTER**  
**(INSTRUKSI PUSH DAN POP)**

**A. TUJUAN**

Dapat menjalankan program aritmatik dengan instruksi PUSH dan POP serta melihat isi dari register SP maupun address yang bersangkutan.

**B. ALAT DAN BAHAN**

1. Lembar tugas
2. Software Z80 Simulator IDE Oshonsoft

**C. TEORI DASAR**

Stack pointer adalah memori tempat simpan isi register 16 bit yang hanya berkaitan dengan operasi penyimpanan PUSH dan operasi pengambilan POP. Register SP (stack pointer) adalah pencatat address stack dan memiliki sifat menunjuk ke address memori lainnya. Operasi register SP selalu berkaitan dnegan operasi PUSH dan POP, di mana operasi ini khusus menyimpan dan mengambil isi register BC, DE, HL, IX, IY dan register AF di stack. Instruksi terseut dapat dijabrkan sebagai berikut:

- Instruksi PUSH qq:

$(SP - 1) \leftarrow qq \text{ (high)}$

$(SP - 2) \leftarrow qq \text{ (low)}$ ; sehingga setelah instruksi PUSH qq maka address register

SP (stack pointer) itu adalah:

$SP \leftarrow SP - 2$

- Instruksi POP qq:

$qq \text{ (low)} \leftarrow (SP)$

$qq \text{ (high)} \leftarrow (SP + 1)$ ; sehingga setelah instruksi POP qq maka address

register SP (stack pointer) itu adalah:

$$SP \leftarrow SP + 2$$

Jadi instruksi PUSH dan POP itu harus digunakan berpasangan. Agar lebih jelas perhatikan program di bawah ini.

LD SP, 1980H : register SP mencatat address 1980, yaitu address awal stack pointer  
LD DE, 1850H : register DE diisi angka 1850H  
LD BC, AAFEH : register BC diisi angka AAFEH  
PUSH DE : simpan isi register DE ke stack  
PUSH BC : simpan isi register BC ke stack  
POP HL : isi stack terakhir di masukkan ke register HL  
POP IX : isi stack terakhir di masukkan ke register IX

Keterangan:

- a. Sebelum dilakukan instruksi PUSH dan POP isi register SP adalah 1980H
- b. Setelah PUSH DE berlangsung, isi register D tersimpan di memori dengan address 197FH dan isi register E tersimpan di memori dengan address 197EH. Register SP menggeser isi catatannya sebanyak 2 kali ke bawah sehingga isi register SP menjadi 197EH.

Jadi hasil dari PUSH DE adalah:

- memori yang addressnya 197FH berisi 18H
  - memori yang addressnya 197EH berisi 50H
  - register SP menjadi berisi 197EH
- c. Setelah PUSH BC berlangsung, isi register B tersimpan di memori dengan address 197DH dan isi register C tersimpan di memori yang addressnya 197CH. Register SP menggeser isi catatannya sebanyak 2 kali ke bawah, sehingga isi register SP menjadi 197CH.

Jadi hasil dari PUSH BC adalah:

- memori yang addressnya 197DH berisi AAH
- memori yang addressnya 197CH berisi FEH
- register SP menjadi berisi 197CH

- d. Setelah POP HL berlangsung, maka isi register L menjadi FEH dan isi register H menjadi AAH. Register SP menggeser isi catatannya 2 kali ke atas, sehingga isi register SP menjadi 197EH. Isi memori yang addressnya 197CH dan 197DH sudah berubah dan bukan FEH maupun AAH.
- register HL menjadi berisi AAFEH
  - register SP menjadi berisi 197EH
  - isi address menjadi 197FE dan 197EH bukan lagi 18H dan 50H
  - isi address memori 197DH dan 197CH bukan lagi AAH dan FEH
- e. Setelah POP IX berlangsung, maka isi register IX menjadi berisi 1850H dan register SP menggeser catatan kembali berisi 1980H. Pada MPF Z-80 register SP tidak harus di-set, karena pada MPF Z-80 register SP sudah di-set oleh program ROM yang ada pada address 1F9FH.

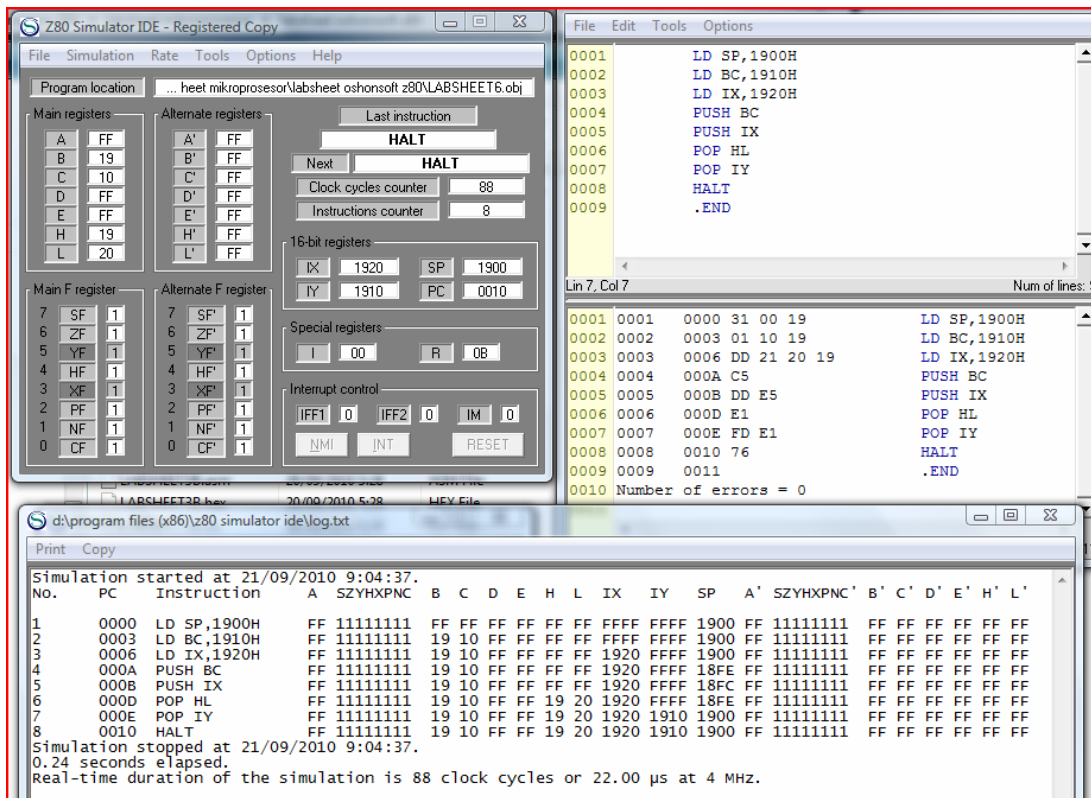
#### **D. LANGKAH KERJA**

##### **PROGRAM I**

1. Jalankan Z80 Simulator IDE
2. Klik Tools / Assembler dan ketik program berikut (tanpa komentar, hanya perintah menmonik) :

LD	SP, 1900H	Set register SP pada 1900H
LD	BC, 1910H	BC ←1910H
LD	IX, 1920H	IX ←1920H
PUSH	BC	SP ←BC
PUSH	IX	SP ←IX
POP	HL	HL ←SP
POP	IY	IY ←SP
HALT		
.END		

3. Pada jendela Assembler klik Tools / Assemble untuk cek kesalahan, simpan file ASM ke folder anda masing-masing.
4. Muat program ke Simulator dengan mengklik Tools / Assemble & Load.
5. Jalankan simulator dengan mengklik Simulation / Start
6. Cek hasil tampilan jendela Simulator Z80, lihat nilai-nilai register apakah sudah sesuai dengan yang diisikan dan cek pula pada tampilan file log.txt



7. Ubah isi register SP = 1820H, BC = 1900H, IX = BCDAH, dan periksa register-register maupun address yang bersangkutan.

## PROGRAM II

1. Jalankan Z80 Simulator IDE
2. Klik Tools / Assembler dan ketik program berikut (tanpa komentar, hanya perintah menmonik) :

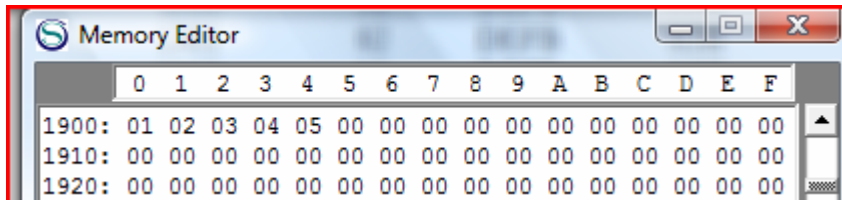
	LD	B, 05H	B ←05H
	LD	HL, 1900H	HL ←1900H
LOOP	PUSH	BC	SP ←BC
	LD	A, (HL)	A ←(HL)
	LD	B, 03H	B ←03H
KALI	ADD	A, A	A ←A + A
	DJNZ	KALI	
	LD	(HL), A	(HL) ← A
	INC	HL	HL ←HL + 1
	POP	BC	BC ←SP
	DJNZ	LOOP	
	HALT		
	.END		

3. Pada jendela Assembler klik Tools / Assemble untuk cek kesalahan, simpan file ASM ke folder anda masing-masing.
4. Muat program ke Simulator dengan mengklik Tools / Assemble & Load.
5. Pada jendela Z80 Simulator IDE klik Tools / Memory Editor untuk memasukkan data di alamat 1900 H sd 1904H dengan cara: scroll layar sampai terlihat lokasi 1900H, kemudian klik pada lokasi tersebut dan masukkan nilai 01 kemudian tekan ENTER , lakukan dengan cara yang sama untuk data sebagai berikut :

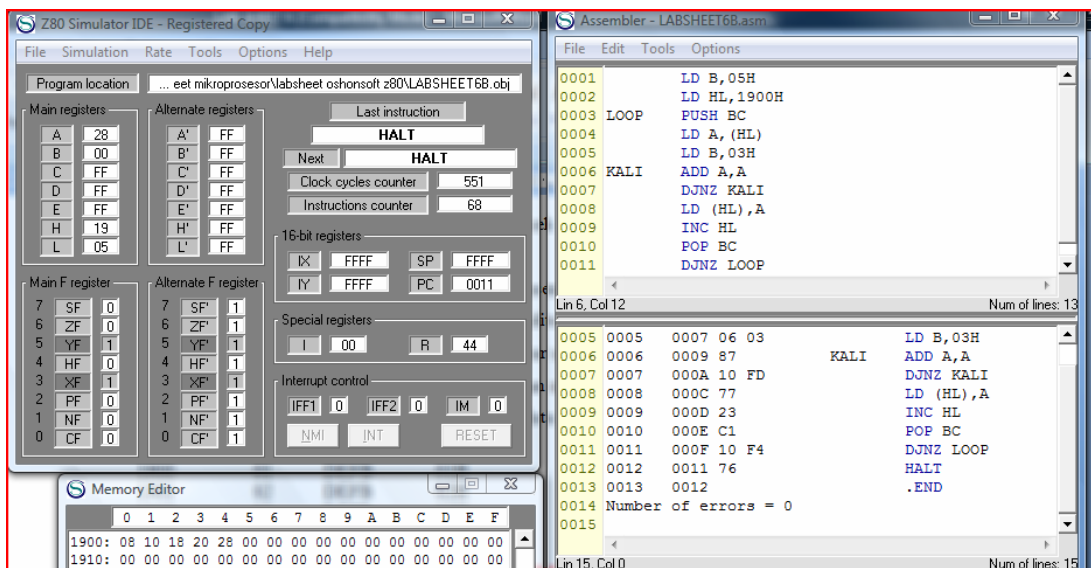
```

ORG 1900H
1900      01      DEFB      01H
1901      02      DEFB      02H
1902      03      DEFB      03H
1903      04      DEFB      04H
1904      05      DEFB      05H

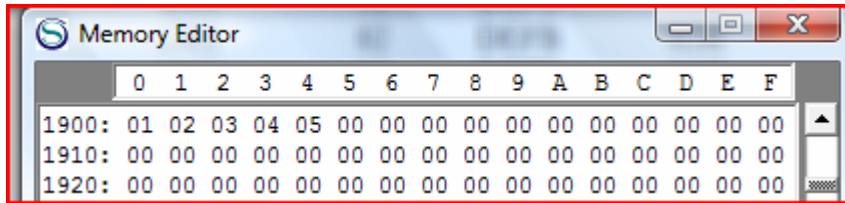
```



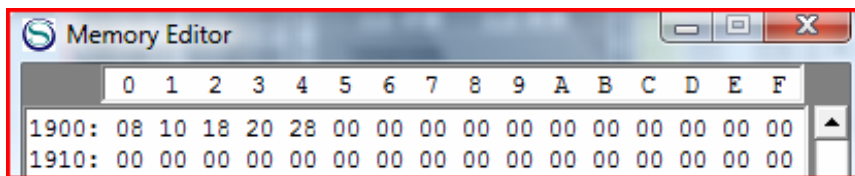
5. Jalankan simulator dengan mengklik Simulation / Start
6. Cek hasil tampilan jendela Simulator Z80, lihat nilai-nilai register apakah sudah sesuai dengan yang diisikan dan cek pula pada tampilan file log.txt



7. Setelah mengetikkan pada MPF-1 maka periksa isi masing-masing register dan address yang terlibat pada program. Terlihat perbedaan isi pada lokasi memori 1900H sd 1904H sebelum dan sesudah program dijalankan.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1900:	01	02	03	04	05	00	00	00	00	00	00	00	00	00	00	00
1910:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1920:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1900:	08	10	18	20	28	00	00	00	00	00	00	00	00	00	00	00
1910:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

8. Buat kesimpulan fungsi dari program diatas. Setelah itu ubah data pada address 1900H sampai dengan 1904H dan jalankan program tersebut dan periksa isi masing-masing register dan address yang terlibat.