

TEKNIK ELEKTRONIKA			
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA			
MIKROPROSESOR	SORTING	LK 05	4X50”

A. TUJUAN

Setelah praktek diharapkan mahasiswa dapat memahami cara kerja dari program sorting dan menerapkannya dengan menggunakan Mikroprosesor Z-80.

B. ALAT DAN BAHAN

Trainer MFP Z-80

C. TEORI SINGKAT

Program sorting adalah suatu program yang dibentuk dengan tujuan untuk mengurutkan data yang akan menghasilkan keluaran data dari yang kecil menuju kebesar. Dalam program sorting, terdapat beberapa instruksi yang digunakan dalam proses pengurutan, seperti :

Instruksi JUMP

Dalam mikroprosesor Zilog Z-80 CPU instruksi – instruksi pencabangan menggunakan instruksi **JUMP**. Instruksi JUMP membuat mikroprosesor menjadi perangkat yang sangat ampuh. Instruksi JUMP dapat dikategorikan menjadi empat kategori, yaitu :

- JUMP bersyarat
- JUMP tanpa syarat
- JUMP absolut
- JUMP relatif

Instruksi Increment dan Decrement

Instruksi INC digunakan untuk menambah isi suatu register atau memori dengan suatu nilai. Instruksi ini sangat potensial digunakan untuk membuat counter cacah naik.

Instruksi DEC digunakan untuk mengurangi isi suatu register dengan suatu memori dengan nilai 1.

Instruksi Compare (CP)

Digunakan untuk membandingkan isi suatu akumulator dengan data immediate 8 bit atau isi salah satu register atau data suatu memori tanpa merubah isi akumulator. Instruksi ini sangat baik untuk menguji sebuah data apakah data tersebut sama dengan nilai tertentu, lebih kecil atau lebih besar.

D. PROCOBAAN

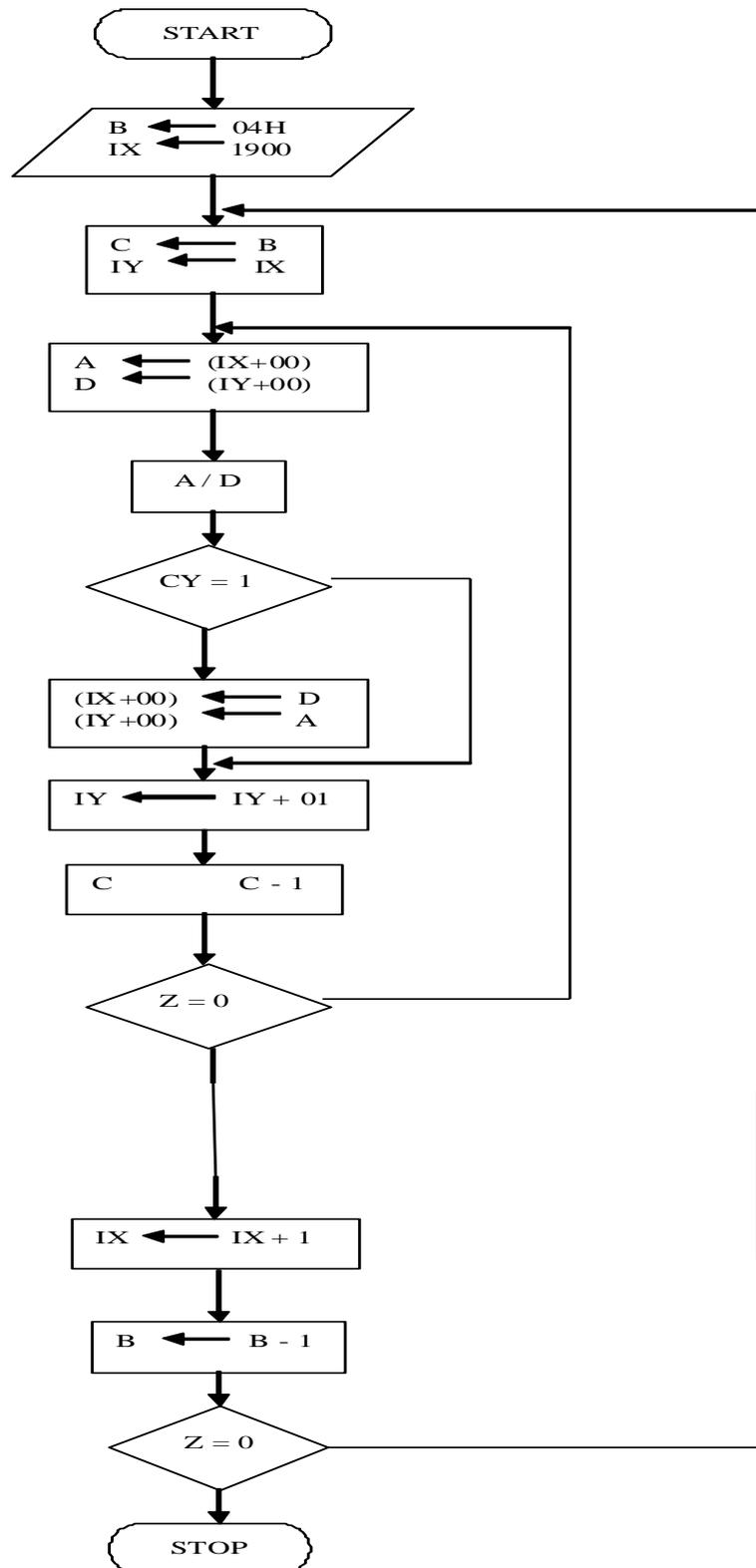
1. Kasus I

Buatlah program untuk mengurutkan data dari yang terkecil menuju yang terbesar.

2. Algoritma

- a. Membuat cacahan jumlah data = 5 di register B
- b. Isi register IX dengan data 1900
- c. Isi register C dengan data pada register B
- d. Isi register IY dengan IX
- e. Isi register A dengan memori yang alamatnya (IX + 00)
- f. Isi register D dengan Memori yang alamatnya (IY + 00)
- g. Bandingkan dengan D
- h. Apakah C = 1, ya lompat ke step 11
- i. Isi memori yang alamatnya (IX + 00) dengan D
- j. Transfer isi reg. D ke memori yang alamatnya (IY + 00)
- k. IY yang baru IY + 00
- l. C yang baru C - 1
- m. Apakah Z = 0, ya ulang ke step 5
- n. IX yang baru IX + 1
- o. B yang baru B - 1
- p. Apakah Z = 0, ya ulang ke step 3
- q. Stop

3. Flow Chart



4. Program

ADDR	Op - Code	Label	No.	Mnemonik
1800	06 04		1	LD B, 04H
1802	DD 21 00 19		2	LD IX, 1900H
1806	48	ULANG	3	LD C,B
1807	DD E5		4	PUSH IX
1809	FD E5		5	PUSH IY
180B	DD 7E 00	ULANG	6	LD A, C (IX + 00)
180E	FD 56 01		7	LD D, C (IY + 01)
1811	ED A9		8	CP D
1813	DA 1C 18		9	JPC, MAJU
1816	DD 72 00		10	LD (IX + 00), D
1819	FD 77 01		11	LD (IY + 01), A
181C	FD 23	MAJU	12	INC IY
181E	0D		13	DEC C
181F	C2 0B 18		14	JPNZ, ULANG
1822	DD 23		15	INC IX
1824	0D		16	DEC B
1825	C2 06 18		17	JPNZ, ULANG
1828	FF		18	RST 38

5. Pengujian

Masukan program adalah data yang nilainya acak, setelah pengekseskusion program maka keluaran program berupa data yang urut dari kecil ke besar yang dapat kita lihat hasilnya pada register AF. Proses pengurutan data dapat dijelaskan dengan ilustrasi berikut :

DATA	I	II	III	IV
80	20 20 20 10	10 10 10	10 10	10
20	80 80 80 80	70 20 20	20 20	20
70	70 70 70 70	80 80 80	70 50	50
10	10 10 10 20	20 70 70	80 80	70
50	50 50 50 50	50 50 50	50 70	80

Cara kerja program :

- Membandingkan data ke-1 dengan ke-2, ke-3 dan data yang lain
- Data yang kecil dijadikan data yang baru untuk dibandingkan
- Setelah dibandingkan dapat yang lebih kecil, data tersebut ditukar lagi
- Program akan berhenti jika semua data telah dibandingkan dan disusun

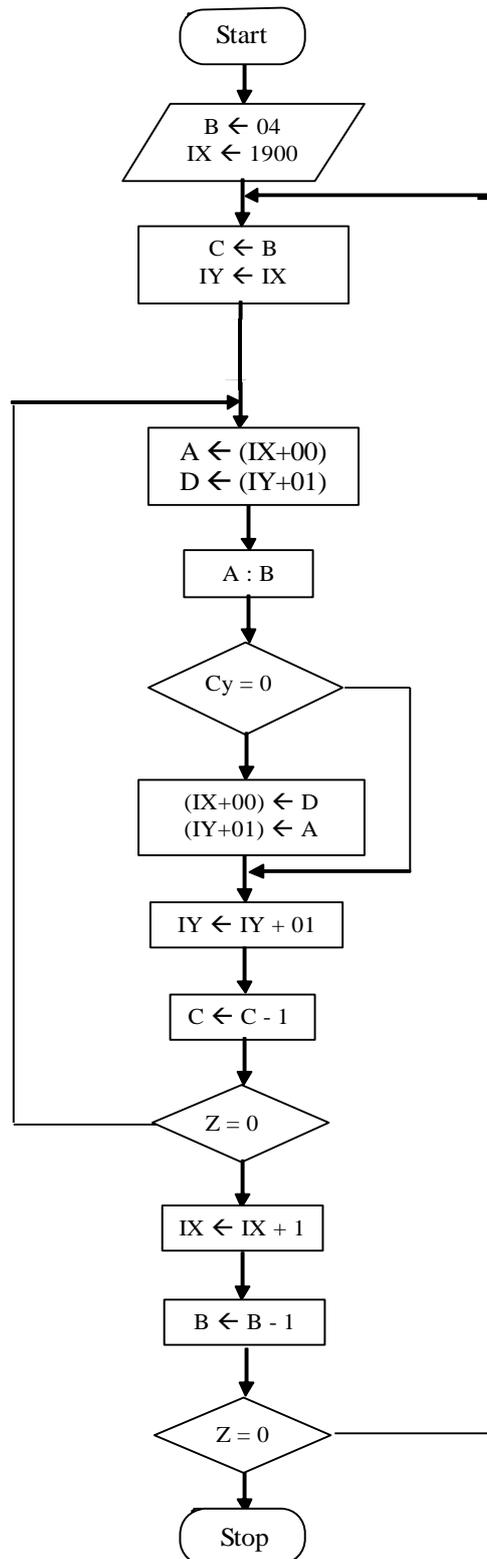
1. Kasus II :

Mengurutkan data dari tertinggi ke terendah

2. Algoritma :

- Buat cacahan jumlah data = 4 di register B
- Set alamat awal tempat simpan data di 1900 di register IX
- Muat data pada register B ke register C
- Muat data pada register IX ke register IY
- Simpan data di memori tempat simpan data register IX ke register A
- Simpan data di memori tempat simpan data register IY +01 ke register D
- Bandingkan isi register A dengan isi register D
- Apakah $Cy = 0$, jika ya lompat ke step j
- Muat data pada register D ke memori tempat simpan data register IX+00
- Naikkan isi register IY dengan 1
- Kurangi isi register C dengan 1
- Apakah $Z = 0$, jika tidak kembali ke step e
- Naikkan isi register IX dengan 1
- Kurangi isi register B dengan 1
- Apakah $Z = 0$, jika tidak kembali ke step c
- Selesai

3. Flow Chart



4. Program

ADD	Op Code	No	Label	Mnemonic	Keterangan
1800	06 04	1		LD B,04	
1802	DD 21 00 19	2		LD IX,1900	
1806	48	3	3	LD C,B	
1807	DD E5	4		PUSH IX	
1809	FD E1	5		POP IY	
180B	DD 7E 00	6	2	LD A,(IX+00)	
180E	FD 56 01	7		LD D,(IY+00)	
180F	BA	8		CP D	
1810	D2 1B 18	9		JPNC,satu	
1813	DD 72 00	10		LD (IX+00),D	
1816	FD 77 01	11		LD (IY+00),A	
1819	FD 23	12	1	INC IY	
181B	0D	13		DEC C	
181C	C2 0B 18	14		JPNZ,dua	
181F	DD 23	15		INC IX	
1821	5	16		DEC B	
1822	C2 06 18	17		JPNZ,tiga	
1825	FF	18		RST 38	

1. Kasus III :

Mencari nilai tertinggi dari beberapa nilai masukan

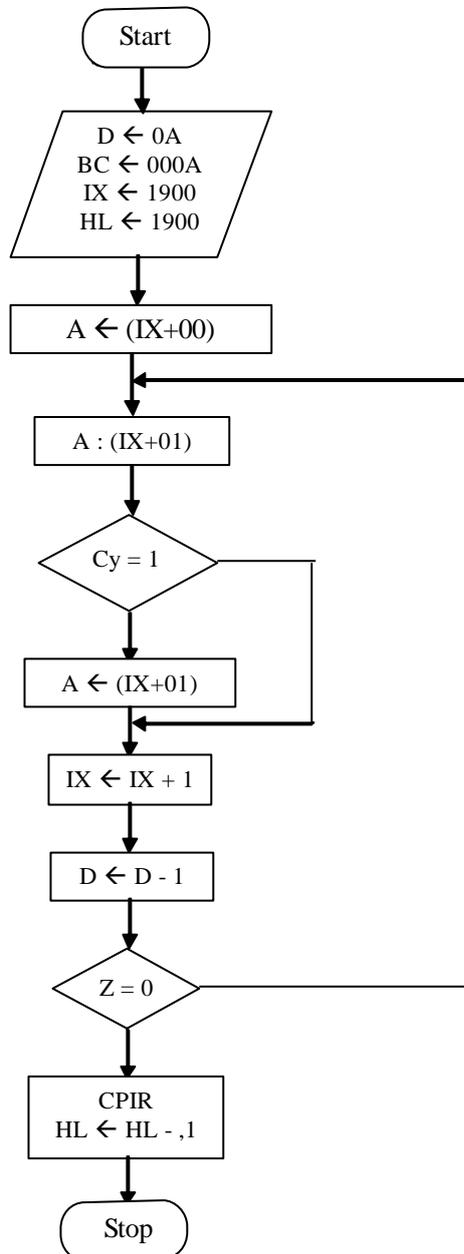
2. Algoritma :

- a. Muat data 0A ke register D
- b. Muat data 000A ke register BC
- c. Set alamat awal tempat simpan data di 1900 di register IX
- d. Set alamat awal tempat simpan data di 1900 di register HL
- e. Simpan data di memori tempat simpan data register IX ke register A
- f. Bandingkan isi memori tempat simpan data register IX+01 dg isi register A
- g. Apakah Cy = 1, jika ya lompat ke step i
- h. Muat data pada memori tempat simpan data register IX+01 ke register A
- i. Naikkan isi register IX dengan 1
- j. Kurangi isi register D dengan 1
- k. Apakah Z = 0, jika tidak kembali ke step f
- l. Bandingkan isi register A dengan data di memori lokasi alamat dicatat HL

m. Kurangi isi register HL dengan 1

n. Selesai

3. Flow Chart



4. Program

ADD	Op Code	No	Label	Mnemonic	Keterangan
1800	16 0A	1		LD D,0A	
1802	01 0A 00	2		LD BC,000A	
1805	DD 21 00 19	3		LD IX,1900	
1809	21 00 19	4		LD HL,1900	
180C	DD 7E 00	5		LD A,(IX+00)	
180F	DD BE 01	6	ulang	CP (IX+01)	
1812	DA 16 18	7		JPC,maju	
1815	DD	8		LD A,(IX+01)	
1816	DD 23	9	maju	INC IX	
1818	15	10		DEC D	
1819	C2 0F 18	11		JPNZ,ulang	
181C	ED B1	12		CPIR	
181E	2B	13		DEC HL	
181F	FF	14		RST 38	