

BAB I

PENDAHULUAN

A. Deskripsi

Modul ini berisi materi latihan kompetensi studi kasus pemrograman sistem mikroprosesor pada Komputer Mikro MPF-I bagi peserta didik SMK program keahlian Elektronika Industri dan Teknik Audio Video.

Melalui modul ini anda dapat berlatih mengembangkan kompetensi mengidentifikasi masalah pemrograman, mendefinisikan masalah pemrograman, menyusun algoritma program, menyusun flowchart program, memilih instruksi yang benar dan tepat, menulis program dan menguji program.

B. Prasyarat

Untuk mempelajari dan melakukan latihan kompetensi dalam modul ini ada dua syarat yaitu :

Syarat Umum :

Anda harus belajar dan berlatih kompetensi dengan rumus “TePUK DisKo” yaitu Teratur, Percaya diri, Ulet, Kreatif, Disiplin dan Konsentrasi.

Syarat Khusus :

Anda harus sudah mempelajari Modul 1 tentang Algoritma program, Modul 2 tentang Arsitektur Komputer Mikro MPF-I, Modul 3 tentang Arsitektur Komputer Mikro MPF-I, Modul 4 Set Instruksi, Sistem digital, Logika, dan memahami matematika Sistem Bilangan Biner, Heksa Desimal

C. Petunjuk Penggunaan Modul

1. Bagi Guru/Fasilitator

- ✓ Baca dan cermati betul deskripsi Silabi
- ✓ Pelajari Kompetensi Dasar, Indikator, Materi Pembelajaran, Kegiatan Pembelajaran, dan Penilaian
- ✓ Pelajari Level Kompetensi Kunci dan Jabarannya
- ✓ Lihat Tujuan Akhir Pembelajaran apakah sudah sesuai dengan Indikator Silabus sebagai tuntutan Kriteria Kinerja deskripsi kompetensi.

- ✓ Cocokkan cakupan kegiatan belajar dengan deskripsi Materi Pembelajaran dan Kegiatan Pembelajaran. Cermati apakah materi kegiatan pembelajaran telah mencakup keseluruhan Kompetensi Dasar dalam aspek Sikap, Pengetahuan, dan Keterampilan.
- ✓ Fasilitasi peserta didik untuk berlatih kompetensi mendefinisikan masalah pemrograman, mengembangkan algoritma pemrograman dan flowchart pemrograman menggunakan bahasa pemrograman Assembly sesuai prinsip pembelajaran KBK yaitu berbasis pada siswa, belajar secara terintegrasi, *Individual learning*, *Mastery learning*, *Problem Solving*, *Experience Based Learning*.

2. Bagi Peserta didik

- ✓ Baca dan pahami deskripsi modul dan prasyarat penggunaan modul
- ✓ Baca dan pahami tujuan akhir modul
- ✓ Isikan rencana kegiatan belajar dengan berkonsultasi dengan guru/fasilitator
- ✓ Baca dan laksanakan cek kemampuan dengan berkonsultasi dengan guru/ fasilitator

D. Tujuan Akhir

Menguasai prosedur pengembangan program

E. KOMPETENSI

BIDANG KEAHLIAN : Teknik Elektronika
 PROGRAM KEAHLIAN : Teknik Elektronika Industri
 STANDAR KOMPETENSI : Memprogram Peralatan Sistem Otomasi Elektronik yang Berkaitan I/O berbantuan : Mikroprosessor dan Mikrokontroler.
 KODE : ELIND 1
 JAM PEMBELAJARAN : 45 (90) Jam @ 45 menit

LEVEL KOMPETENSI	A	B	C	D	E	F	G
KUNCI	3	3	3	3	2	3	3

KONDISI KINERJA	Unjuk kerja ketrampilan kognitif namun dengan imajinasi psiko-motorik seperti unit kompetensi ini bisa dicapai dengan kondisi: <ol style="list-style-type: none"> 1. Memiliki kemampuan dasar tentang konsep sistem 2. Memiliki kompetensi dasar elektronika 3. Memiliki kemampuan mengenai petunjuk keselamatan kerja secara umum 4. Memiliki kemampuan menulis laporan kerja yang baik.
-----------------	---

KOMPETENSI DASAR	INDIKATOR	MATERI PEMBELAJARAN	KEGIATAN PEMBELAJARAN	PENILAIAN	ALOKASI WAKTU			SUMBER BELAJAR
					TM	PS	PI	
1. Menguasai Prosedur Penyusunan Algoritma Pemrograman	1.1. Ditunjukkan kaidah-kaidah yang berlaku dalam penyusunan algoritma dan pemrograman.	Kaidah-kaidah penyusunan algoritma	<ul style="list-style-type: none"> ▪ Memahami pengertian Program Komputer ▪ Menguraikan empat langkah Pengembangan Program Komputer ▪ Berlatih mendefinisikan Permasalahan Program ▪ Berlatih mengembangkan Algoritma Program ▪ Berlatih memilih bentuk baku Flowchart ▪ Berlatih menyusun Flowchart 	<ul style="list-style-type: none"> • Tes Lisan • Tes Tulis 	10	10 (20)	-	<i>Microprocessor and Interfacing Programming and Hardware</i> MCGraw-Hill, 1992

KOMPETENSI DASAR	INDIKATOR	MATERI PEMBELAJARAN	KEGIATAN PEMBELAJARAN	PENILAIAN	ALOKASI WAKTU			SUMBER BELAJAR
					TM	PS	PI	
	1.2. Diujikan algoritma dan pemrograman yang telah dikuasai dengan melakukan beberapa kasus.	Pengujian algoritma dan pemrograman	<ul style="list-style-type: none"> ▪ Menguji algoritma dan pemrograman ▪ Menggunakan algoritma dalam pemrograman ▪ Menguji studi kasus penggunaan algoritma dalam pemrograman ▪ Memahami Bahasa Pemrograman Koputer 	<ul style="list-style-type: none"> • Tes Lesan • Tes Tulis 	10	5 (10)	10 (40)	<i>Microprocessor and Interfacing Programming and Hardware</i> MCGraw-Hill, 1992

LEVEL KOMPETENSI KUNCI

No	Kompetensi Kunci	LEVEL 1	LEVEL 2	LEVEL 3
A	MENGUMPULKAN, MENGANALISA DAN MENGELOLA INFORMASI Kapasitas untuk mengumpulkan informasi, memindahkan dan menyeleksi informasi dalam rangka memilih informasi yang diperlukan untuk dipresentasikan, mengevaluasi sumber dan cara memperoleh informasi tersebut	Mengakses dan menyimpan dari satu sumber	Mengakses, memilih dan menyimpan dari beberapa sumber	<i>Mengakses, mengevaluasi dan mengatur dari berbagai macam sumber</i>
B	MENKOMUNIKASIKAN IDE-IDE DAN INFORMASI Kapasitas untuk berkomunikasi dengan orang lain secara efektif menggunakan beragam bahasa, tulisan, grafik dan ekspresi non verbal lainnya	Sederhana dengan aturan yang telah dikenal	Komplek dengan isi tertentu	<i>Komplek dengan isi beragam</i>
C	MERENCANAKAN DAN MENGORGANISIR KEGIATAN Kapasitas untuk merencanakan dan mengatur kegiatan kerja individu termasuk penggunaan waktu dan sumber yang baik, pemilihan prioritas dan pengawasan prestasi individu	Di bawah pengawasan	Dengan bimbingan	<i>Inisiatif sendiri dan mengevaluasi kegiatan yang kompleks</i>

No	Kompetensi Kunci	LEVEL 1	LEVEL 2	LEVEL 3
D	BEKERJA DENGAN ORANG LAIN SERTA KELOMPOK DALAM SATU TIM Kapasitas untuk berhubungan secara efektif dengan orang lain baik antar pribadi ataupun kelompok termasuk mengerti dan memberikan respon akan keinginan klien dan bekerja secara efektif sebagai anggota kelompok untuk mencapai tujuan bersama.	Aktifitas yang telah diketahui	Membantu merancang dan mencapai tujuan	<i>Kolaborasi dalam kegiatan kelompok</i>
E	MENGGUNAKAN IDE-IDE SERTA TEKNIK DALAM MATEMATIKA Kapasitas untuk menggunakan konsep bilangan, spasi dan ukuran dan teknik seperti perkiraan untuk praktek	Tugas yang sederhana	<i>Memilih tugas yang kompleks dan sesuai</i>	Evaluasi dan mengadaptasi sebagai tugas yang sesuai
F	MENYELESAIKAN MASALAH Kapasitas untuk menjalankan strategi penyelesaian masalah baik untuk situasi di mana masalah dan solusi yang diinginkan memiliki bukti dan dalam situasi yang memerlukan pemikiran kritis dan pendekatan kreatif untuk mencapai hasil	Rutin, sedikit pengawasan eksplorasi – pengawasan melekat	Rutin, mandiri eksplorasi- dengan bimbingan	<i>Masalah yang kompleks, pelaksanaan pendekatan sistematis, menjelaskan proses</i>
G	MENGGUNAKAN TEKNOLOGI Kapasitas untuk menerangkan teknologi, mengkombinasikan keahlian fisik dan sensor yang diperlukan untuk menjalankan peralatan dengan pengertian ilmiah dan prinsip teknologi yang diperlukan untuk mengadaptasi sistem	Produksi ulang atau melaksanakan produk dasar atau jasa	Menyusun, mengatur atau mengoperasikan produk atau jasa	<i>Mendesain atau merangkai produk atau jasa</i>

F. Cek Kemampuan

Berilah tanda thick (✓) pada kolom YA atau TIDAK sesuai dengan pernyataan berikut.

Mintalah Catatan dan Tanda Tangan Guru/Pembimbing.

NO	PERNYATAAN	YA	TIDAK	CATATAN GURU/PEMBIMBING	TANDA TANGAN GURU/PEMBIMBING
01	Apakah saudara sudah memahami cara mengembangkan program pada mikroprosesor				
02	Apakah saudara sudah memahami cara menulis program				
03	Apakah saudara sudah sering menggunakan Komputer mikro MPF-I untuk membuat program				

BAB II PEMBELAJARAN

A. Rencana Belajar Peserta didik

Rencana belajar peserta didik diisi oleh peserta didik dan disetujui oleh Guru. Rencana belajar tersebut adalah sebagai berikut :

NAMA PESERTA DIDIK : _____

JENIS KEGIATAN	TANGGAL	WAKTU	TEMPAT BELAJAR	ALASAN PERUBAHAN	TANDA TANGAN GURU

B. Kegiatan Belajar

1. Kegiatan Belajar I: Percobaan Transfer Data

a. TUJUAN :

- 1) Memahami pemakaian fungsi instruksi transfer data
- 2) Menentukan nilai seting inisialisasi data
- 3) Menulis program
- 4) Memasukkan program (op-code) ke Komputer Mikro MPF-I
- 5) Mengeksekusi program
- 6) Menganalisis hasil eksekusi

b. PERALATAN YANG DIPERLUKAN

Trainer MPF-1

c. PERCOBAAN

1). Percobaan I

Kasus :

Tuliskan program dalam bahasa assembly untuk mengeset isi register-register A=00h, B=01h, C=02h, D=03h, E=04h, H=05h, L=06h dengan menggunakan instruksi transfer data 8 bit.

Penyelesaian:

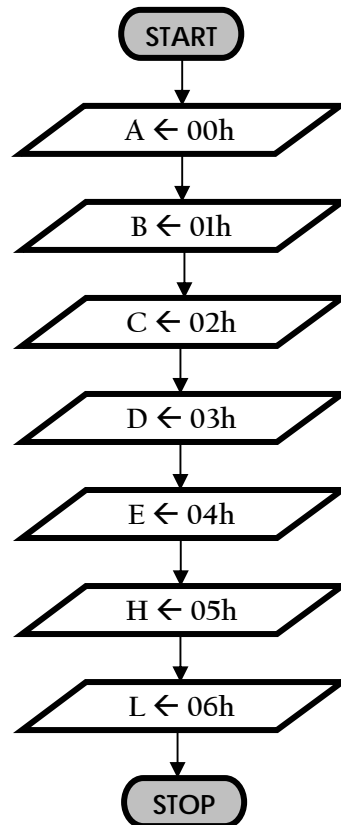
a). Algoritma :

Langkah pertama buatlah algoritma yang menunjukkan step atau langkah kerja proses kerja program sebagai berikut.

STEP	PERINTAH
1.	Start
2.	Isi register A dengan data 00h
3.	Isi register B dengan data 01h
4.	Isi register C dengan data 02h
5.	Isi register D dengan data 03h
6.	Isi register E dengan data 04h
7.	Isi register H dengan data 05h
8.	Isi register L dengan data 06h
9.	Stop

b). Flowchart:

Langkah kedua yang harus dilakukan adalah membuat flowchart sebagai diagram yang menunjukkan proses aliran kerja program. Perhatikan pemilihan bentuk-bentuk grafis flowchart harus benar karena akan mewakili kesesuaian pemilihan jenis dan macam instruksi.



c). Penulisan Program :

Penulisan program dalam bahasa assembly pertama-tama disarankan membuat tabel dengan format seperti dibawah ini. Setiap baris isilah nomor 1 sampai dengan ke n. Perhatikan perintah program mulai dari yang ke satu, termasuk kategori apa lalu tetapkan perintah assembly-nya. Tuliskan semua pada kolom Assembly. Cari kode operasi masing-masing perintah menggunakan tabel set instruksi tuliskan ke kolom Kode Operasi. Kemudian berikan alamat sesuai jumbuh byte kode operasinya.

Addr	Kode Operasi	No.	Label	Assembly	Keterangan
1800	3E 00	1.		LD A , 00H	
1802	06 01	2.		LD B , 01H	
1804	0E 02	3.		LD C , 02H	
1806	16 03	4.		LD D , 03H	
1808	1E 04	5.		LD E , 04H	
180A	26 05	6.		LD H , 05H	
180C	2E 06	7.		LD L , 06H	
180E	FF	8.		RST 38	

d). Pemasukan Program dan Eksekusi Program

- Masukkan semua kode operasi program sesuai alamat setiap perintah.
- Cek kembali apakah semua kode operasi telah benar dan sesuai dengan posisi alamatnya.
- Jika semua sudah benar lanjutkan dengan operasi eksekusi menggunakan eksekusi GO dan baca hasil program di masing-masing register dengan mencatat pada tabel berikut :

Register	A	B	C	D	E	H	L
Data							

- Bagaimana dengan hasil eksekusi program saudara apakah sesuai dengan kasus/ permasalahan. Jika tidak sesuai berarti masih ada kesalahan yang harus dicari dan diperbaiki.

2). Percobaan 2

Kasus :

Tuliskan program dalam bahasa assembly untuk mengeset isi register-register B=BBh, C=CCh, D=DDh, E=EEh, H=FFh, L=00h dengan menggunakan instruksi transfer data 16 bit.

Penyelesaian:

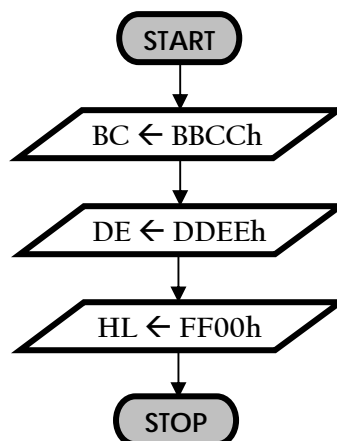
a). Algoritma :

Langkah pertama buatlah algoritma yang menunjukkan step atau langkah kerja proses kerja program sebagai berikut.

STEP	PERINTAH
1.	Start
2.	Isi register BC dengan data BBCCh
3.	Isi register DE dengan data DDEEh
4.	Isi register HL dengan data FF00h
5.	Stop

b). Flowchart:

Langkah kedua yang harus dilakukan adalah membuat flowchart sebagai diagram yang menunjukkan proses aliran kerja program. Perhatikan pemilihan bentuk-bentuk grafis flowchart harus benar karena akan mewakili kesesuaian pemilihan jenis dan macam instruksi.



c). **Penulisan Program :**

Penulisan program dalam bahasa assembly pertama-tama disarankan membuat tabel dengan format seperti dibawah ini. Setiap baris isilah nomor 1 sampai dengan ke n. Perhatikan perintah program mulai dari yang ke satu, termasuk kategori apa lalu tetapkan perintah assembly-nya. Tuliskan semua pada kolom Assembly. Cari kode operasi masing-masing perintah menggunakan tabel set instruksi tuliskan ke kolom Kode Operasi. Kemudian berikan alamat sesuai jumbuh byte kode operasinya.

Addr	Kode Operasi	No.	Label	Assembly	Keterangan
1800	01 CC DD	1.		LD BC , BBCCH	
1802	11 EE DD	2.		LD DE , DDEEH	
1804	21 00 FF	3.		LD HL, FF00H	
180E	FF	4.		RST 38	

d). **Pemasukan Program dan Eksekusi Program**

- 1). Masukkan semua kode operasi program sesuai alamat setiap perintah.
- 2). Cek kembali apakah semua kode operasi telah benar dan sesuai dengan posisi alamatnya.
- 3). Jika semua sudah benar lanjutkan dengan operasi eksekusi menggunakan eksekusi GO dan baca hasil program di masing-masing register dengan mencatat pada tabel berikut :

Register	BC	DE	HL
Data			

- 4). Bagaimana dengan hasil eksekusi program saudara apakah sesuai dengan kasus/ permasalahan. Jika tidak sesuai berarti masih ada kesalahan yang harus dicari dan diperbaiki.

3). Percobaan III

Kasus :

Tuliskan program dalam bahasa assembly untuk menghapus 50 lokasi isi memori pada alamat 1900H sampai 1931H dengan data 00H.

Penyelesaian:

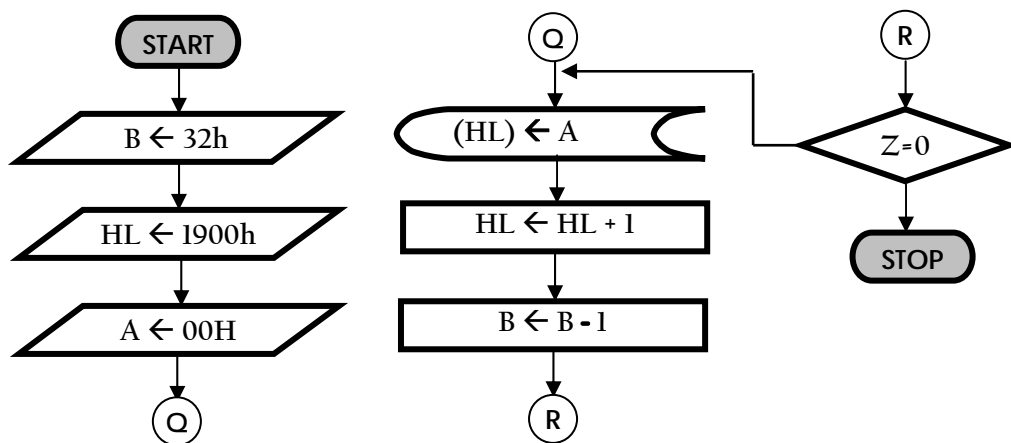
a). Algoritma :

Langkah pertama buatlah algoritma yang menunjukkan step atau langkah kerja proses kerja program sebagai berikut.

STEP	PERINTAH
1.	Start
2.	Isi register B dengan data 32h = 50 desimal
3.	Isi register HL dengan data 1900h
4.	Isi register A dengan data 00h
5.	Isi memori pada alamat sama dengan isi register HL dengan data register A
6.	Tambahkan nilai isi register HL dengan satu
7.	Kurangkan isi register B dengan satu
8.	Apakah isi register BC tidak sama dengan Nol
9.	Jika Ya kembali ke Step 5
10.	Stop

b). Flowchart:

Langkah kedua yang harus dilakukan adalah membuat flowchart sebagai diagram yang menunjukkan proses aliran kerja program. Perhatikan pemilihan bentuk-bentuk grafis flowchart harus benar karena akan mewakili kesesuaian pemilihan jenis dan macam instruksi.



c). **Penulisan Program :**

Penulisan program dalam bahasa assembly pertama-tama disarankan membuat tabel dengan format seperti dibawah ini. Setiap baris isilah nomor 1 sampai dengan ke n. Perhatikan perintah program mulai dari yang ke satu, termasuk kategori apa lalu tetapkan perintah assembly-nya. Tuliskan semua pada kolom Assembly. Cari kode operasi masing-masing perintah menggunakan tabel set instruksi tuliskan ke kolom Kode Operasi. Kemudian berikan alamat sesuai jumbuh byte kode operasinya.

Addr	Kode Operasi	No.	Label	Assembly	Keterangan
1800	06 32	1.		LD B , 32H	
1802	21 00 19	2.		LD HL , 1900H	
1805	AF	3.		XOR A	A = 00
1806	77	4.	ULANG	LD (HL), A	
1807	23	5.		INC HL	
1808	05	6.		DEC B	
1809	C2 06 18	7.		JPNZ, ULANG	Loop ke 1806
180C	FF	8.		RST 38	

d). **Pemasukan Program dan Eksekusi Program**

- 1). Masukkan semua kode operasi program sesuai alamat setiap perintah.
- 2). Cek kembali apakah semua kode operasi telah benar dan sesuai dengan posisi alamatnya.
- 3). Jika semua sudah benar lanjutkan dengan operasi eksekusi menggunakan eksekusi GO dan baca hasil program di memori mulai alamat 1900 sampai dengan alamat 1931 dengan mencatat pada tabel berikut :

Memori	1900	1901	1902	1903	1904	dst	1931
Data							

- 4). Bagaimana dengan hasil eksekusi program saudara apakah sesuai dengan kasus/ permasalahan. Jika tidak sesuai berarti masih ada kesalahan yang harus dicari dan diperbaiki.

4). Percobaan IV

Kasus :

Tuliskan program dalam bahasa assembly untuk mengeset data : 00h, 01h, 02h, 03h, 03h, 04h, 05h, 06h, 07h, 08h, 09h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, dan 0Fh di memori RWM mulai alamat 1900 sampai 190F.

Penyelesaian:

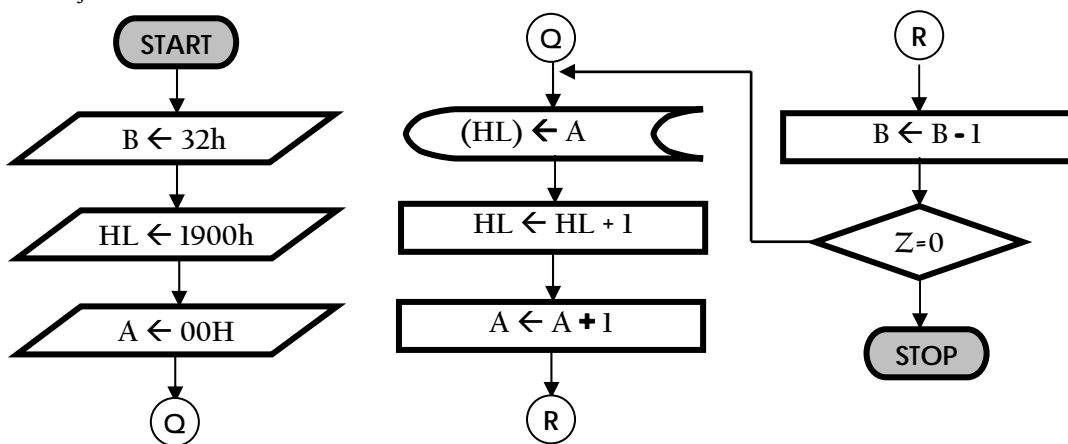
a). Algoritma :

Langkah pertama buatlah algoritma yang menunjukkan step atau langkah kerja proses kerja program sebagai berikut.

STEP	PERINTAH
1.	Start
2.	Isi register B dengan data 10h = 16 desimal
3.	Isi register HL dengan data 1900h
4.	Isi register A dengan data 00h
5.	Isi memori pada alamat sama dengan isi register HL dengan data register A
6.	Tambahkan nilai/isi register HL dengan satu
7.	Tambahkan nilai/isi register A dengan satu
8.	Kurangkan nilai/isi register B dengan satu
9.	Apakah isi register BC tidak sama dengan Nol
10.	Jika Ya kembali ke Step 5
11.	Stop

b). Flowchart:

Langkah kedua yang harus dilakukan adalah membuat flowchart sebagai diagram yang menunjukkan proses aliran kerja program. Perhatikan pemilihan bentuk-bentuk grafis flowchart harus benar karena akan mewakili kesesuaian pemilihan jenis dan macam instruksi.



c). **Penulisan Program :**

Penulisan program dalam bahasa assembly pertama-tama disarankan membuat tabel dengan format seperti dibawah ini. Setiap baris isilah nomor 1 sampai dengan ke n. Perhatikan perintah program mulai dari yang ke satu, termasuk kategori apa lalu tetapkan perintah assembly-nya. Tuliskan semua pada kolom Assembly. Cari kode operasi masing-masing perintah menggunakan tabel set instruksi tuliskan ke kolom Kode Operasi. Kemudian berikan alamat sesuai jumbuh byte kode operasinya.

Addr	Kode Operasi	No.	Label	Assembly	Keterangan
1800	06 32	1.		LD B , 32H	
1802	21 00 19	2.		LD HL , 1900H	
1805	AF	3.		XOR A	A = 00
1806	77	4.	ULANG	LD (HL), A	
1807	23	5.		INC HL	
1808	3C	6.		INC A	
1809	05	7.		DEC B	
180A	C2 06 18	8.		JPNZ, ULANG	Loop ke 1806
180D	FF	9.		RST 38	

d). **Pemasukan Program dan Eksekusi Program**

- 1). Masukkan semua kode operasi program sesuai alamat setiap perintah.
- 2). Cek kembali apakah semua kode operasi telah benar dan sesuai dengan posisi alamatnya.
- 3). Jika semua sudah benar lanjutkan dengan operasi eksekusi menggunakan eksekusi GO dan baca hasil program di memori mulai alamat 1900 sampai dengan alamat 190F dengan mencatat pada tabel berikut :

Memori	1900	1901	1902	1903	1904	1905	1906	1907
Data								

Memori	1908	1909	190A	190B	190C	190D	190E	190F
Data								

- 4). Bagaimana dengan hasil eksekusi program saudara apakah sesuai dengan kasus/permasalahan. Jika tidak sesuai berarti masih ada kesalahan yang harus dicari dan diperbaiki

2. Kegiatan Belajar II: Percobaan Aplikasi Aritmetika “Jumlah-Kurang”

a. TUJUAN :

- 1) Memahami pemakaian fungsi instruksi aritmetika
- 2) Menentukan nilai seting inisialisasi data
- 3) Menulis program
- 4) Memasukkan program (op-code) ke Komputer Mikro MPF-I
- 5) Mengeksekusi program
- 6) Menganalisis hasil eksekusi

b. PERALATAN YANG DIPERLUKAN

Trainer MPF-1

c. PERCOBAAN

1). Percobaan I

Kasus.

Buatlah Algoritma Program, Flow Chart dan Program untuk menjumlahkan 2 byte bilangan biner yang masing-masing beralamat di 1900H, 1901H, 1902H, 1903H. Simpanlah hasilnya di alamat 1904H, 1905H, 1906H. Program di tulis pada alamat awal program 1800H.

Pola masalah :

$$\begin{array}{r} [1901] [1900] \\ [1903] [1902] + \\ \hline [1906] [1905] [1904] \end{array}$$

Penyelesaian:

a). Algoritma :

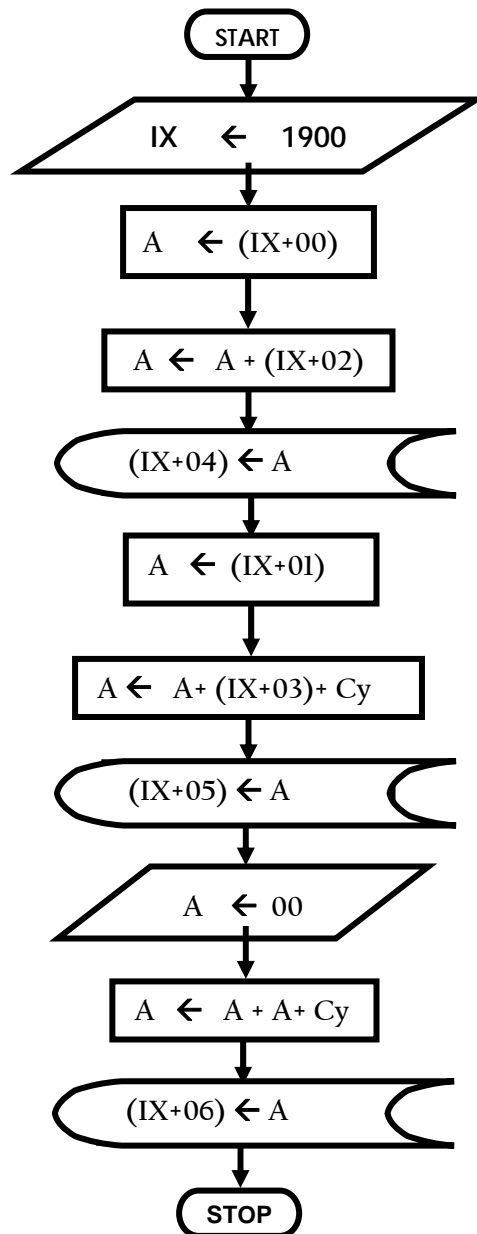
Langkah pertama buatlah algoritma yang menunjukkan step atau langkah kerja proses kerja program sebagai berikut.

STEP	PERINTAH
1.	Start
2.	Ambil data yang ada pada alamat 1900 dan simpan di Reg. A
3.	Jumlahkan data yang ada di register A dengan data yang ada pada alamat 1902
4.	Simpan hasil penjumlahan yang ada di register A pada alamat 1904
5.	Ambil data yang ada pada alamat 1901 dan simpan di Reg. A

STEP	PERINTAH
6.	Jumlahkan data yang ada di register A dengan data yang ada pada alamat 1903 dan sertakan bit Carry Flag
7.	Simpan hasil penjumlahan yang ada di register A pada alamat 1905
8.	Ambil bit Carry Flag
9.	Simpan pada alamat 1906
10.	Stop

b). Flowchart:

Langkah kedua yang harus dilakukan adalah membuat flowchart sebagai diagram yang menunjukkan proses aliran kerja program.



c). Penulisan Program :

Penulisan program dalam bahasa assembly pertama-tama disarankan membuat tabel dengan format seperti dibawah ini. Setiap baris isilah nomor 1 sampai dengan ke n. Perhatikan perintah program mulai dari yang ke satu, termasuk kategori apa lalu tetapkan perintah assembly-nya. Tuliskan semua pada kolom Assembly. Cari kode operasi masing-masing perintah menggunakan tabel set instruksi tuliskan ke kolom Kode Operasi. Kemudian berikan alamat sesuai jumbuh byte kode operasinya.

Addr	Kode Operasi	No.	Label	Assembly	Keterangan
1800	DD 21 00 19	1		LD IX , 1900	
1804	DD 7E 00	2		LD A , (IX+00)	
1807	DD 86 02	3		ADD A , (IX+02)	
180A	DD 77 04	4		LD (IX+04) , A	
180D	DD 7E 01	5		LD A , (IX+01)	
1810	DD 8E 03	6		ADC A , (IX+03)	
1813	DD 77 05	7		LD (IX+05) , A	
1816	3E 00	8		LD A , 00	
1818	8F	9		ADC A , A	
1819	DD 77 06	10		LD (IX+06) , A	
181C	FF	11		RST 38	

d). Pemasukan Program dan Eksekusi Program :

- Masukkan program di atas dan uji menggunakan data seperti tabel berikut ini;
- Sebelum dieksekusi masukkan data 1 dan data 2 di memori sesuai alamat satu persatu dari kasus 1 sampai dengan kasus 5.

Kasus	Data 1		Data 2		Data 1 + Data 2		
	[1900]	[1901]	[1902]	[1903]	[1904]	[1905]	[1906]
1	10	00	03	02
2	06	05	08	07
3	8F	9F	F5	AF
4	2C	7F	77	2D
5	F4	2A	7C	C2

e). Bagaimana dengan hasil eksekusi program saudara apakah sesuai dengan kasus/ permasalahan. Jika tidak sesuai berarti masih ada kesalahan yang harus dicari dan diperbaiki. Konsultasilah dengan Guru atau Tutor saudara jika ada masalah.

2). Percobaan II

Kasus.

Buatlah Algoritma Program, Flow Chart dan Program untuk menjumlahkan 2 byte bilangan desimal yang masing-masing beralamat di 1900H, 1901H, 1902H, 1903H. Simpanlah hasilnya di alamat 1904H, 1905H, 1906H. Program di tulis pada alamat awal program 1800H.

Pola masalah :

$$\begin{array}{r} [1901] [1900] \\ [1903] [1902] + \\ \hline [1906] [1905] [1904] \end{array}$$

Penyelesaian:

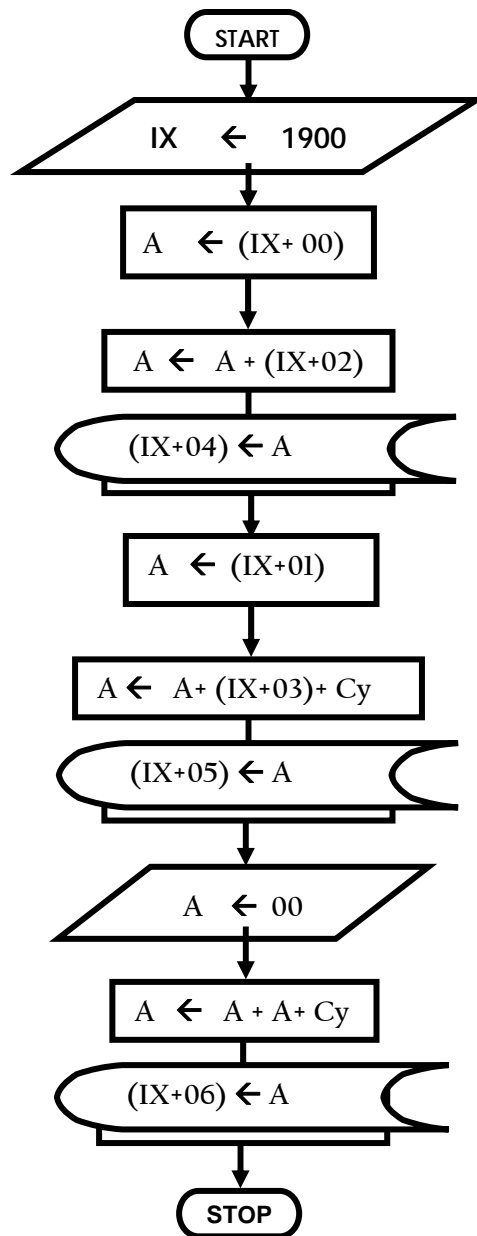
a). Algoritma :

Langkah pertama buatlah algoritma yang menunjukkan step atau langkah kerja proses kerja program sebagai berikut.

STEP	PERINTAH
1.	Start
2.	Ambil data yang ada pada alamat 1900 dan simpan di Reg. A
3.	Jumlahkan data yang ada di register A dengan data yang ada pada alamat 1902
4.	Lakukan koreksi DAA
5.	Simpan hasil penjumlahan yang ada di register A pada alamat 1904
6.	Ambil data yang ada pada alamat 1901 dan simpan di Reg. A
7.	Jumlahkan data yang ada di register A dengan data yang ada pada alamat 1903 dan sertakan bit Carry Flag
8.	Lakukan koreksi DAA
9.	Simpan hasil penjumlahan yang ada di register A pada alamat 1905
10.	Ambil bit Carry Flag
11.	Simpan pada alamat 1906
12.	Stop

b). Flowchart:

Langkah kedua yang harus dilakukan adalah membuat flowchart sebagai diagram yang menunjukkan proses aliran kerja program.



c). Penulisan Program :

Penulisan program dalam bahasa assembly pertama-tama disarankan membuat tabel dengan format seperti dibawah ini. Setiap baris isilah nomor 1 sampai dengan ke n. Perhatikan perintah program mulai dari yang ke satu, termasuk kategori apa lalu tetapkan perintah assembly-nya. Tuliskan semua pada kolom Assembly. Cari kode operasi masing-masing perintah menggunakan tabel set instruksi tuliskan ke kolom Kode Operasi. Kemudian berikan alamat sesuai jumbuh byte kode operasinya.

Addr	Kode Operasi	No.	Label	Assembly	Keterangan
1800	DD 21 00 19	1.		LD IX , 1900	
1804	DD 7E 00	2.		LD A , (IX+00)	
1807	DD 86 02	3.		ADD A , (IX+02)	
180A	27	4.		DAA	Koreksi
180B	DD 77 04	5.		LD (IX+04) , A	
180E	DD 7E 01	6.		LD A , (IX+01)	
1811	DD 8E 03	7.		ADC A , (IX+03)	
1814	27	8.		DAA	Koreksi
1815	DD 77 05	9.		LD (IX+05) , A	
1818	3E 00	10.		LD A , 00	
181A	8F	11.		ADC A , A	
181B	DD 77 06	12.		LD (IX+06) , A	
181E	FF	13.		RST 38	

d). Pemasukan Program dan dan Eksekusi Program :

- Masukkan program di atas dan uji menggunakan data seperti tabel berikut ini;
- Sebelum dieksekusi masukkan data 1 dan data 2 di memori sesuai alamat satu persatu dari kasus 1 sampai dengan kasus 5.

Kasus	Data 1		Data 2		Data 1 + Data 2		
	[1900]	[1901]	[1902]	[1903]	[1904]	[1905]	[1906]
1	10	00	03	02
2	06	05	08	07
3	45	90	25	15
4	90	80	77	65
5	64	77	29	21

e). Bagaimana dengan hasil eksekusi program saudara apakah sesuai dengan kasus/ permasalahan. Jika tidak sesuai berarti masih ada kesalahan yang harus dicari dan diperbaiki. Konsultasilah dengan Guru atau Tutor saudara jika ada masalah.

3). Percobaan III

Kasus.

Buatlah Algoritma Program, Flow Chart dan Program untuk pengurangan 1 byte bilangan biner yang masing-masing beralamat di 1900H, 1901H. Simpanlah hasilnya di alamat 1902. Program di tulis pada alamat awal program 1800H.

Pola masalah :

$$\begin{array}{r}
 [1900] \\
 [1901] - \\
 \hline
 [1902]
 \end{array}$$

Penyelesaian:

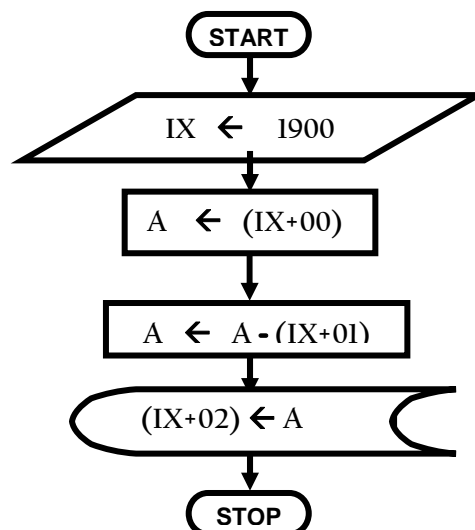
a). Algoritma :

Langkah pertama buatlah algoritma yang menunjukkan step atau langkah kerja proses kerja program sebagai berikut.

STEP	PERINTAH
1.	Start
2.	Ambil data yang ada pada alamat 1900 dan simpan di Reg. A
3.	Kurangkan data yang ada di register A dengan data yang ada pada alamat 1901
4.	Simpan hasil pengurangan yang ada di register A pada alamat 1902
5.	Stop

b). Flowchart:

Langkah kedua yang harus dilakukan adalah membuat flowchart sebagai diagram yang menunjukkan proses aliran kerja program.



c) Penulisan Program :

Addr	Kode Operasi	No.	Label	Assembly	Keterangan
1800	DD 21 00 19	1.		LD IX , 1900	
1804	DD 7E 00	2.		LD A , (IX+00)	
1807	DD 96 01	3.		SUB , (IX+01)	
180A	DD 77 02	4.		LD (IX+02) , A	
180D	FF	5.		RST 38	

d). Pengujian Program :

Masukkan program di atas dan uji menggunakan data seperti tabel berikut :

Kasus	[1900]	[1901]	[1902]
1	F5	2D
2	E7	D5
3	8F	F5

Program diatas adalah program pengurangan bilangan biner. Untuk membuat program tersebut bekerja sebagai program pengurangan bilangan desimal maka perlu di tambahkan satu perintah koreksi DAA seperti program berikut :

Addr	Kode Operasi	No.	Label	Assembly	Keterangan
1800	DD 21 00 19	1		LD IX , 1900	
1804	DD 7E 00	2		LD A , (IX+00)	
1807	DD 96 01	3		SUB , (IX+01)	
180A	27	4		DAA	
180B	DD 77 02	5		LD (IX+02) , A	
180E	FF	6		RST 38	

Pengujian Program :

Masukkan program di atas dan uji menggunakan data seperti tabel berikut :

Kasus	[1900]	[1901]	[1902]
1	80	40
2	90	25
3	55	15

Program pengurangan dapat dibuat menjadi program penjumlahan dengan menegatifkan bilangan pengurangnya. Program berikut menghasilkan keluaran yang sama.

Addr	Kode Operasi	No.	Label	Assembly	Keterangan
1800	DD 21 00 19	1		LD IX , 1900	
1804	DD 7E 01	2		LD A , (IX+01)	
1807	ED 44	3		NEG	
1809	47	4		LD B , A	
180A	DD 7E 00	5		LD A , (IX+00)	
180D	80	6		ADD , B	
180E	27	7		DAA	
180F	DD 77 02	8		LD (IX+02) , A	
1812	FF	9		RST 38	

Pengujian Program :

Masukkan program di atas dan uji menggunakan data seperti tabel berikut :

[1900]	[1901]	[1902]
80	40	
90	25	
55	15	

3. Kegiatan Belajar III : Percobaan Aplikasi Aritmetika “Kali-Bagi”

Saudara tentu masih ingat bahwa Mikroprosesor Z-80 CPU tidak memiliki instruksi aritmetika perkalian. Operasi perkalian biner tak bertanda dapat dilakukan dengan dua cara yaitu : (1) dengan menggeser bilangan biner ke kiri dan (2) melakukan penjumlahan berulang.

Contoh :

$$\begin{array}{r} 0101 = 5d \\ \times 1011 = 11d \\ \hline 0101 \\ 0101 \\ 0000 \\ + 0101 \\ \hline 011011 = 55d \end{array}$$

Cara Geser

$$\begin{aligned} 0101 \times 1011 &= 0101 + 0101 + 0101 + 0101 + 0101 + \\ &0101 + 0101 + 0101 + 0101 + 0101 + \\ &0101 \\ &= 011011 \end{aligned}$$

Cara Jumlah berulang

Perkalian cara geser bekerja memeriksa bilangan pengali. Jika bit tersebut berlogika 1, bilangan yang dikalikan ditulis sama. Jika bit tersebut adalah 0, tuliskan 0000. Posisi bilangan-bilangan yang disimpan tersebut diatur sehingga bit berorde paling kecil dari bilangan yang dikalikan berada satu garis dengan bit bilangan pengali yang sedang diperiksa. Dalam contoh diatas, bilangan yang dikalikan dengan bilangan pengali adalah 4 bit. Jadi operasi pemeriksaan, penggeseran, dan penjumlahan diulang 4 kali. Untuk perkalian 8 bit operasi diulang 8 kali, dan untuk perkalian 16 bit operasi diulang 16 kali.

Perkalian cara penjumlahan berulang, bilangan yang dikalikan dijumlahkan sebanyak bilangan pengalinya.

4). Percobaan IV

a. Kasus :

Kalikan data 8 bit pada register E dengan data 8 bit pada register A. Hasilnya disimpan pada register HL.

Penyelesaian: Cara I dengan Metoda Geser Bit

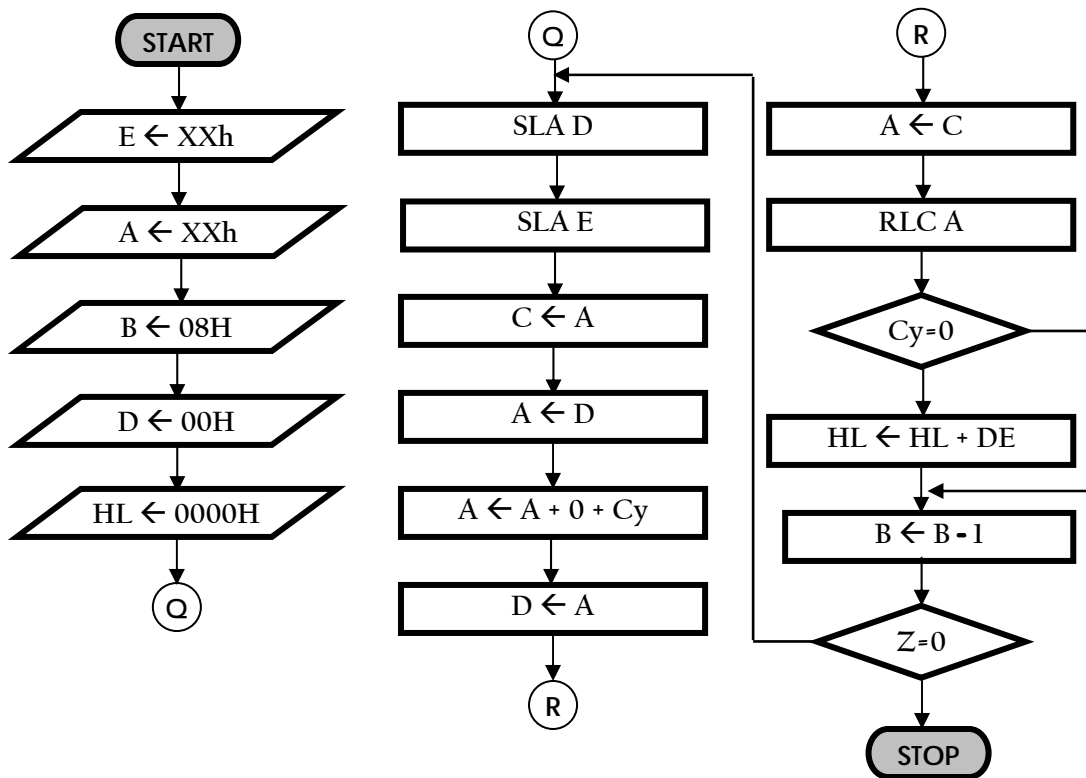
a). Algoritma :

Langkah pertama buatlah algoritma yang menunjukkan step atau langkah kerja proses kerja program sebagai berikut.

STEP	PERINTAH
1.	Start
2.	Set data pada Register E sebagai data Bilangan yang dikalikan
3.	Set data pada Register A sebagai data Bilangan pengali
4.	Set Register B = 8 sebagai pengulangan 8 kali loop
5.	Set Register D = 00h
6.	Set Register HL = 0000h
7.	Geser kekiri isi Register D (SLA D)
8.	Geser ke kiri isi Register E (SLA E)
9.	Jumlahkan isi Register D dengan 0 sertakan bit Cy
10.	Putar bit Data register A ke kiri (RLC A)
11.	Apakah Cy = 0
12.	Jika Ya lompat ke STEP 14
13.	Jumlahkan isi Register HL dengan isi Register DE
14.	Kurangi isi Register B dengan 1
15.	Apakah Z = 0
16.	Jika Ya kembali ke STEP 7
17.	Stop

b). Flowchart:

Langkah kedua yang harus dilakukan adalah membuat flowchart sebagai diagram yang menunjukkan proses aliran kerja program.



c). Penulisan Program :

Penulisan program dalam bahasa assembly pertama-tama disarankan membuat tabel dengan format seperti dibawah ini. Setiap baris isilah nomor 1 sampai dengan ke n. Perhatikan perintah program mulai dari yang ke satu, termasuk kategori apa lalu tetapkan perintah assembly-nya. Tuliskan semua pada kolom Assembly. Cari kode operasi masing-masing perintah menggunakan tabel set instruksi tuliskan ke kolom Kode Operasi. Kemudian berikan alamat sesuai jumbuh byte kode operasinya.

Addr	Kode Operasi	No.	Label	Assembly	Keterangan
1800	1E XX	1.		LD E , XX	bil. dikalikan
1802	3E XX	2.		LD A , XX	bil. pengali
1804	06 08	3.		LD B, 08H	Σ putaran
1806	16 00	4.		LD D, 00H	
1808	21 00 00	5.		LD HL, 0000H	
180B	CB 22	6.	LOOP	SLA D	
180D	CB 23	7.		SLA E	
180F	4F	8.		LD C, A	
1810	7A	9.		LD A, D	
1811	CE 00	10.		ADC A, 00H	
1813	57	11.		LD D, A	
1814	79	12.		LD A, C	

Addr	Kode Operasi	No.	Label	Assembly	Keterangan
1815	07	13.		RLCA	
1816	D2 1A 18	14.		JP NC, MAJU	
1819	19	15.		ADD HL, DE	
181A	10 EF	16.	MAJU	DJNZ, LOOP	
181C	FF	17.		RST 38	

d). Pemasukan Program dan Eksekusi Program :

- Masukkan program di atas dan uji menggunakan data seperti tabel berikut ini;
- Sebelum dieksekusi masukkan nilai Register A dan Register E dengan mengganti Op-Code XX sesuai data kasus di bawah ini

Kasus	Data 1	Data 2	Data 1 X Data 2
	Register A	Register E	Register HL
1	25	05	
2	45	09	
3	2C	0A	
4	F5	09	
5	D2	0F	

e). Bagaimana dengan hasil eksekusi program saudara apakah sesuai dengan kasus/ permasalahan. Jika tidak sesuai berarti masih ada kesalahan yang harus dicari dan diperbaiki. Konsultasilah dengan Guru atau Tutor saudara jika ada masalah.

f). Program diatas dapat dibuat lebih sederhana seperti program berikut ini.

Addr	Kode Operasi	No.	Label	Assembly	Keterangan
1800	1E XX	1.		LD E , XX	bil. dikalikan
1802	3E XX	2.		LD A , XX	bil. pengali
1804	06 08	3.		LD B, 08H	Σ putaran
1806	16 00	4.		LD D, 00H	
1808	21 00 00	5.		LD HL, 0000H	
180B	29	6.	LOOP	ADD HL,HL	
180C	07	7.		RLCA	
180D	30 01	8.		JR NC, MAJU	
180F	19	9.		ADD HL,DE	
1810	10 F9	10.	MAJU	DJN, LOOP	
1812	FF	11.		RST 38	STOP

Penyelesaian: Cara II dengan Metoda Penjumlahan Berulang

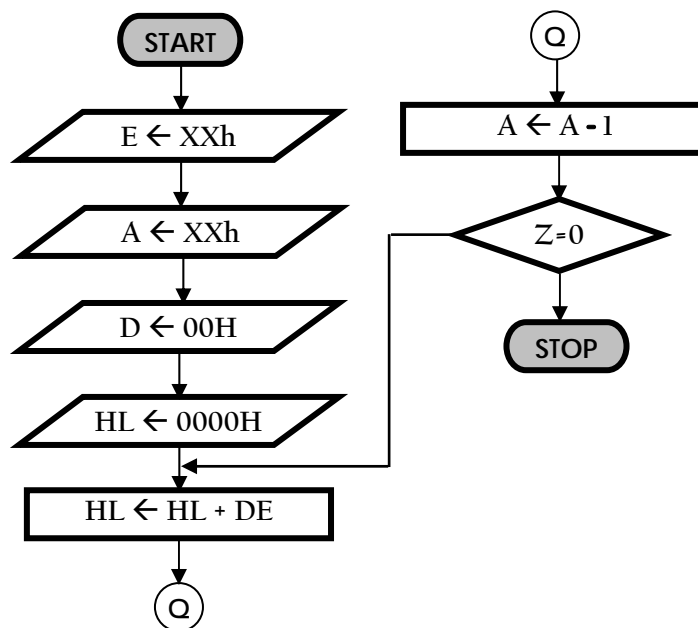
a). Algoritma :

Langkah pertama buatlah algoritma yang menunjukkan step atau langkah kerja proses kerja program sebagai berikut.

STEP	PERINTAH
1.	Start
2.	Set data pada Register E sebagai data Bilangan yang dikalikan
3.	Set data pada Register A sebagai data Bilangan pengali
4.	Set Register D = 00h
5.	Set Register HL = 0000h
6.	Jumlahkan isi Register HL dengan isi Register DE
7.	Kurangi A dengan 1
8.	Apakah Z = 0
9.	Jika Ya kembali ke STEP 6
10.	Stop

b). Flowchart:

Langkah kedua yang harus dilakukan adalah membuat flowchart sebagai diagram yang menunjukkan proses aliran kerja program.



c). **Penulisan Program :**

Penulisan program dalam bahasa assembly pertama-tama disarankan membuat tabel dengan format seperti dibawah ini. Setiap baris isilah nomor 1 sampai dengan ke n. Perhatikan perintah program mulai dari yang ke satu, termasuk kategori apa lalu tetapkan perintah assembly-nya. Tuliskan semua pada kolom Assembly. Cari kode operasi masing-masing perintah menggunakan tabel set instruksi tuliskan ke kolom Kode Operasi. Kemudian berikan alamat sesuai jumbuh byte kode operasinya.

Addr	Kode Operasi	No.	Label	Assembly	Keterangan
1800	1E XX	1.		LD E , XX	bil. dikalikan
1802	3E XX	2.		LD A , XX	bil. pengali
1804	16 00	3.		LD D, 00H	Σ putaran
1806	21 00 00	4.		LD HL, 0000H	
1809	19	5.	LOOP	ADD HL, DE	
180A	3D	6.		DEC A	
180B	C2 09 18	7.		JP NZ, LOOP	
181E	FF	8.		RST 38	

d). **Pemasukan Program dan Eksekusi Program :**

- Masukkan program di atas dan uji menggunakan data seperti tabel berikut ini;
- Sebelum dieksekusi masukkan nilai Register A dan Register E dengan mengganti Op-Code XX sesuai data kasus di bawah ini

Kasus	Data 1	Data 2	Data 1 X Data 2
	Register A	Register E	Register HL
1	25	05	
2	45	09	
3	2C	0A	
4	F5	09	
5	D2	0F	

e). Bagaimana dengan hasil eksekusi program saudara apakah sesuai dengan kasus/ permasalahan. Jika tidak sesuai berarti masih ada kesalahan yang harus dicari dan diperbaiki. Konsultasilah dengan Guru atau Tutor saudara jika ada masalah.

5). Percobaan V

a. Kasus :

Bagi data 8 bit pada register A dengan data 8 bit pada register E. Hasilnya disimpan pada register C.

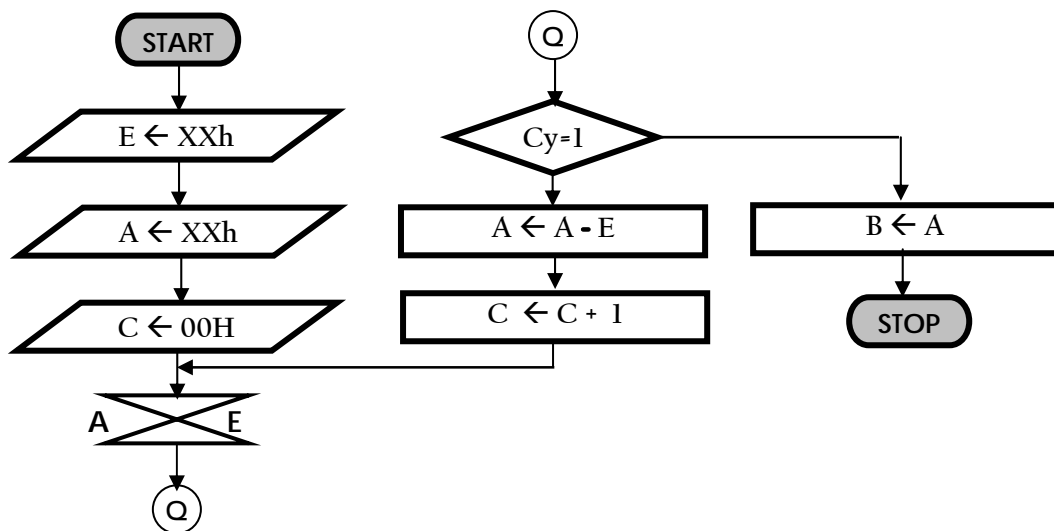
a). Algoritma :

Langkah pertama buatlah algoritma yang menunjukkan step atau langkah kerja proses kerja program sebagai berikut.

STEP	PERINTAH
1.	Start
2.	Set data pada Register E sebagai data Bilangan pembagi
3.	Set data pada Register A sebagai data Bilangan yang dibagi
4.	Set Register C = 00h
5.	Bandingkan isi Register A dengan isi Register E
6.	Apakah Cy = 1
7.	Jika Ya lompat ke STEP 11
8.	Kurangkan isi register A dengan isi register E
9.	Tambahkan isi register C dengan 1
10.	Kembali ke STEP 5
11.	Copykan isi register A ke register B
12.	Stop

b). Flowchart:

Langkah kedua yang harus dilakukan adalah membuat flowchart sebagai diagram yang menunjukkan proses aliran kerja program.



c). Penulisan Program :

Penulisan program dalam bahasa assembly pertama-tama disarankan membuat tabel dengan format seperti dibawah ini. Setiap baris isilah nomor 1 sampai dengan ke n. Perhatikan perintah program mulai dari yang ke satu, termasuk kategori apa lalu tetapkan perintah assembly-nya. Tuliskan semua pada kolom Assembly. Cari kode operasi masing-masing perintah menggunakan tabel set instruksi tuliskan ke kolom Kode Operasi. Kemudian berikan alamat sesuai jumbuh byte kode operasinya.

Addr	Kode Operasi	No.	Label	Assembly	Keterangan
1800	1E XX	1.		LD E , XX	bil. pembagi
1802	3E XX	2.		LD A , XX	bil. terbagi
1804	0E 00	3.		LD C, 00H	
1806	BB	4.	LOOP	CP, E	
1807	DA OE 18	5.		JP C, HENTI	
1809	93	6.		SUB E	
180A	0C	7.		INC C	
180B	C3 06 18	8.		JP, LOOP	
189E	47	9.	HENTI	LD B, A	
181F	FF	10.		RST 38	

d). Pemasukan Program dan Eksekusi Program :

- Masukkan program di atas dan uji menggunakan data seperti tabel berikut ini;
- Sebelum dieksekusi masukkan nilai Register A dan Register E dengan mengganti Op-Code XX sesuai data kasus di bawah ini

Kasus	Data 1	Data 2	Data 1 : Data 2	
	Register A	Register E	Register B	Register B
1	25	05		
2	45	09		
3	2C	0A		
4	F5	09		
5	D2	0A		

e). Bagaimana dengan hasil eksekusi program saudara apakah sesuai dengan kasus/ permasalahan. Jika tidak sesuai berarti masih ada kesalahan yang harus dicari dan diperbaiki. Konsultasilah dengan Guru atau Tutor saudara jika ada masalah.

4. Kegiatan Belajar IV: Tes Formatif

- a. Buatlah Algoritma Program, Flow Chart dan Program untuk membangkitkan 100 data bilangan desimal mulai dari 0 di memori mulai alamat 1900.
- b. Buatlah Algoritma Program, Flow Chart dan Program untuk membangkitkan 50 data bilangan desimal ganjil mulai dari 1 di memori mulai alamat 1900.
- c. Buatlah Algoritma Program, Flow Chart dan Program untuk membangkitkan 50 data bilangan desimal genap mulai dari 2 di memori mulai alamat 1900.
- d. Buatlah Algoritma Program, Flow Chart dan Program untuk mencacah jumlah data bernilai 99 dari 100 data yang ada di memori mulai alamat 1900. Hasil cacahan dicatat di register C.
- e. Buatlah Algoritma Program, Flow Chart dan Program untuk mencacah jumlah data bernilai 90 sampai dengan 95 dari 100 data yang ada di memori mulai alamat 1900. Hasil cacahan dicatat di register C.
- f. Buatlah Algoritma Program, Flow Chart dan Program untuk mencacah jumlah data bernilai lebih kecil dari 90 dari 100 data yang ada di memori mulai alamat 1900. Hasil cacahan dicatat di register C.
- g. Buatlah Algoritma Program, Flow Chart dan Program untuk mengkonversi 50 bilangan Desimal menjadi bilangan Heksa/Biner. Bilangan desimal tercatat di memori mulai alamat 1900 dan hasil konversi bilangan biner tercatat di memori mulai alamat 1A00.
- h. Buatlah Algoritma Program, Flow Chart dan Program untuk mengurutkan 40 data dari kecil ke besar di memori mulai alamat 1900.
- i. Buatlah Algoritma Program, Flow Chart dan Program untuk mengurutkan 40 data dari besar ke kecil di memori mulai alamat 1900.

KUNCI JAWABAN

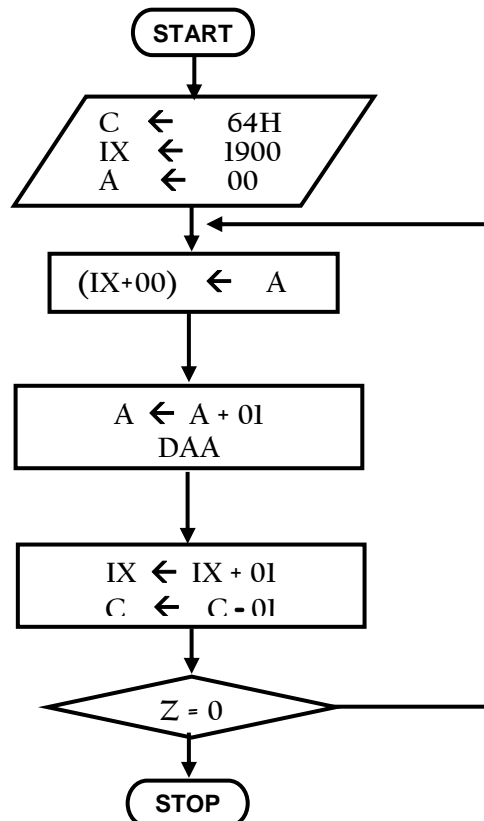
Penyelesaian Kasus a

Membangkitkan 100 data desimal dari 00 s/d 99 di memori alamat 1900 – 1963

Algoritma:

STEP	PERINTAH
1.	Start
2.	Buat cacahan data sama dengan 100d (64h) di register C
3.	Set alamat awal tempat simpan data 1900 di register IX
4.	Set data awal sama dengan 0 di register A
5.	Simpan data register A di memori yang alamatnya di catat oleh register IX
6.	Jumlahkan data pada register A dengan 1
7.	Desimalkan
8.	Naikkan nilai pencatat alamat register IX satu alamat
9.	Kurangkan isi register C dengan 1
10.	Apakah nilai Z = 0
11.	Jika Ya kembali ke STEP 5
12.	Stop

Flow Chart



Program :

ADDR	Kode Operasi	No.	Label	Assembly	Keterangan
1800	0E 64	1		LD C , 64H	
1802	DD 21 00 19	2		LD IX , 1900H	
1806	3E 00	3		LD A , 00H	
1808	DD 77 00	4	Catat	LD (IX+00) , A	
180B	C6 01	5		ADD A , 01	
180D	27	6		DAA	
180E	DD 23	7		INC IX	
1810	0D	8		DEC C	
1811	C2 08 18	9		JP NZ, Catat	
1814	FF	10		RST 38	

Pengujian Program :

Masukkan program di atas lalu eksekusi dan baca memori alamat 1900 sampai alamat 1963 akan tercatat data desimal 00 sampai dengan 99.

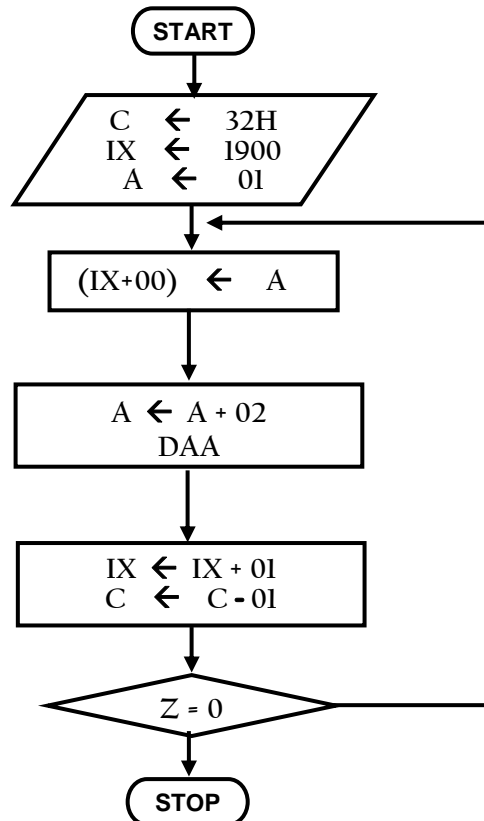
Penyelesaian Kasus b.

Program untuk membangkitkan 50 data bilangan desimal ganjil mulai dari 1 di memori mulai alamat 1900

Algoritma:

STEP	PERINTAH
1.	Start
2.	Buat cacahan data sama dengan 50d (32h) di register C
3.	Set alamat awal tempat simpan data 1900 di register IX
4.	Set data awal sama dengan 1 di register A
5.	Simpan data register A di memori yang alamatnya di catat oleh register IX
6.	Jumlahkan data pada register A dengan 2
7.	Desimalkan
8.	Naikkan nilai pencatat alamat register IX satu alamat
9.	Kurangkan isi register C dengan 1
10.	Apakah nilai Z = 0
11.	Jika Ya kembali ke STEP 5
12.	Stop

Flow Chart



Program :

ADDR	Kode Operasi	No.	Label	Assembly	Keterangan
1800	0E 32	1		LD C , 32H	
1802	DD 21 00 19	2		LD IX , 1900H	
1806	3E 01	3		LD A , 01H	
1808	DD 77 00	4	Catat	LD (IX+00) , A	
180B	C6 02	5		ADD A , 02	
180D	27	6		DAA	
180E	DD 23	7		INC IX	
1810	0D	8		DEC C	
1811	C2 08 18	9		JP NZ, Catat	
1814	FF	10		RST 38	

Pengujian Program :

Masukkan program di atas lalu eksekusi dan baca memori alamat 1900 sampai alamat 1931 akan tercatat data desimal ganjil 01 sampai dengan 99.

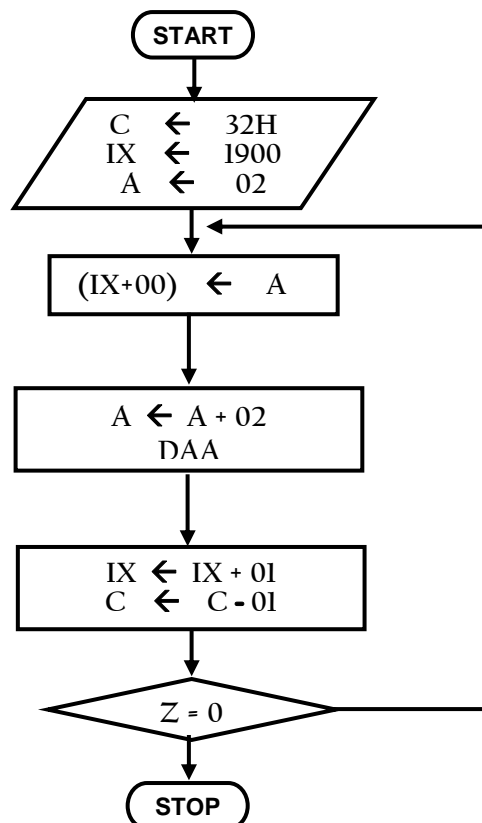
Penyelesaian Kasus c

Program untuk membangkitkan 50 data bilangan desimal genap mulai dari 2 di memori mulai alamat 1900.

Algoritma:

STEP	PERINTAH
1.	Start
2.	Buat cacahan data sama dengan 50d (32h) di register C
3.	Set alamat awal tempat simpan data 1900 di register IX
4.	Set data awal sama dengan 2 di register A
5.	Simpan data register A di memori yang alamatnya di catat oleh register IX
6.	Jumlahkan data pada register A dengan 2
7.	Desimalkan
8.	Naikkan nilai pencatat alamat register IX satu alamat
9.	Kurangkan isi register C dengan 1
10.	Apakah nilai Z = 0
11.	Jika Ya kembali ke STEP 5
12.	Stop

Flow Chart



Program :

ADDR	Kode Operasi	No.	Label	Assembly	Keterangan
1800	0E 32	1		LD C , 32H	
1802	DD 21 00 19	2		LD IX , 1900H	
1806	3E 02	3		LD A , 02H	
1808	DD 77 00	4	Catat	LD (IX+00) , A	
180B	C6 02	5		ADD A , 02	
180D	27	6		DAA	
180E	DD 23	7		INC IX	
1810	0D	8		DEC C	
1811	C2 08 18	9		JP NZ, Catat	
1814	FF	10		RST 38	

Pengujian Program :

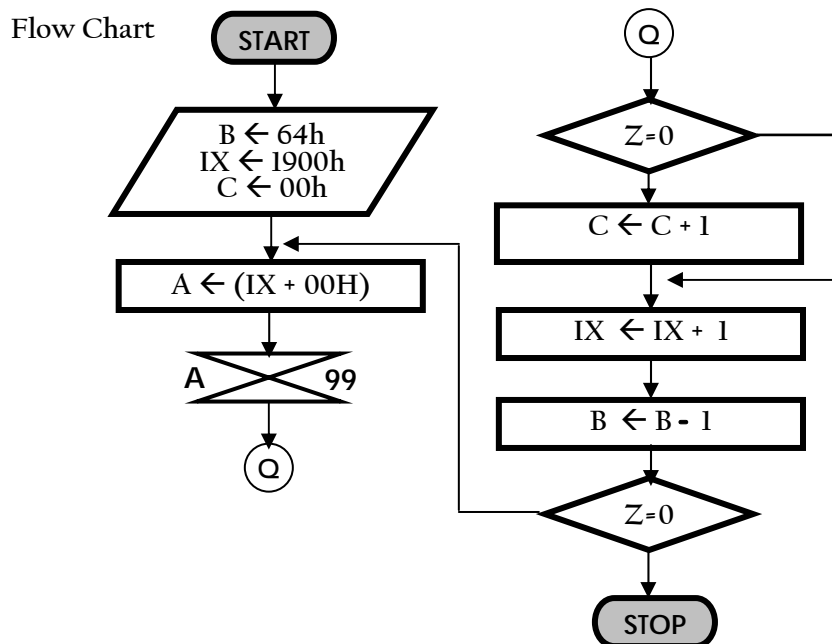
Masukkan program di atas lalu eksekusi dan baca memori alamat 1900 sampai alamat 1963 akan tercatat data desimal genap dari 02 sampai dengan 98.

Penyelesaian Kasus d

Program untuk mencacah jumlah data bernilai 99 dari 100 data yang ada di memori mulai alamat 1900.

Algoritma:

STEP	PERINTAH
1.	Start
2.	Buat cacahan data sama dengan 100d (64h) di register B
3.	Set alamat awal tempat simpan data 1900 di register IX
4.	Set pencatat cacahan register C = 0
5.	Ambil data dari memori yang alamatnya di catat oleh reg. IX simpan di register A
6.	Bandingkan data pada register A dengan 99
7.	Apakah Z = 0
8.	Jika Ya lompat ke STEP 10
9.	Tambahkan nilai cacahan register C dengan 1
10.	Naikkan nilai pencatat alamat register IX satu alamat
11.	Kurangkan isi register B dengan 1
12.	Apakah nilai Z = 0
13.	Jika Ya kembali ke STEP 5
14.	Stop



Program :

ADDR	Kode Operasi	No.	Label	Assembly	Keterangan
1800	06 64	1		LD B , 64H	
1802	DD 21 00 19	2		LD IX , 1900H	
1806	0E 00	3		LD C , 00H	
1808	DD 7E 00	4	ULANG	LD A, (IX+00)	
180B	FE 99	5		CP, 99	
180D	C2 11 18	6		JP NZ, MAJU	
1810	0C	7		INC C	
1811	DD 23	8	MAJU	INC IX	
1813	10 F3	9		DJNZ, ULANG	
1815	FF	10		RST 38	

Pengujian Program :

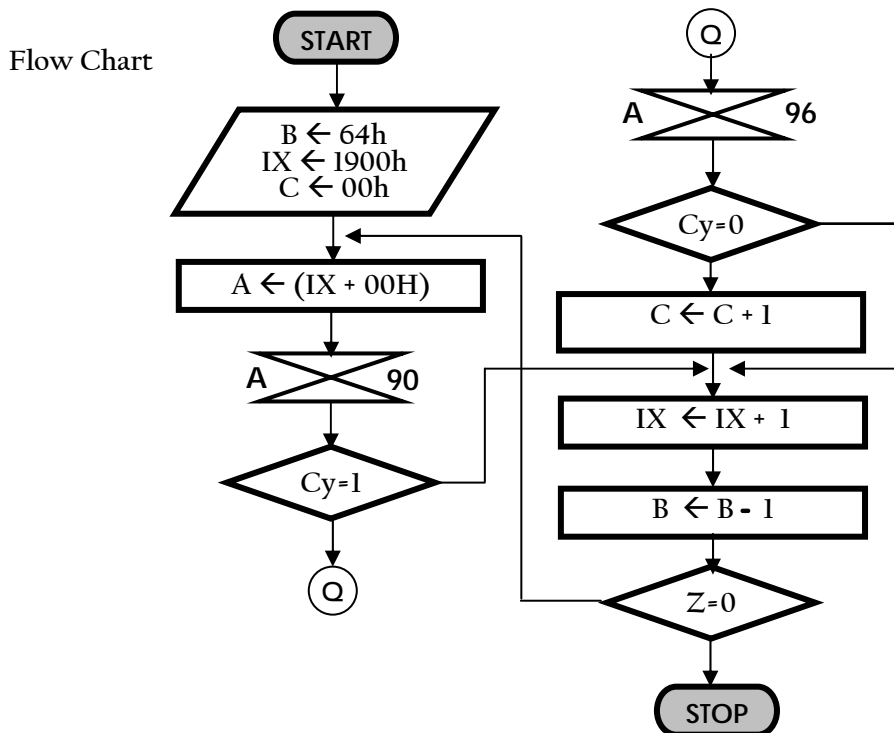
Masukkan program di atas dan masukkan 100 data secara acak mulai alamat 1900 lalu eksekusi dan baca isi register C. Apakah isi register C menunjukkan jumlah cacahan data bernilai 99 ?. Jika ya berarti program saudara sudah benar. Jika tidak cermati sekali lagi apakah masih ada kesalahan. Jangan lupa bertanya pada tutor saudara.

Penyelesaian Kasus e

Program untuk mencacah jumlah data bernilai 90 sampai dengan 95 dari 100 data yang ada di memori mulai alamat 1900. Hasil cacahan dicatat di register C.

Algoritma:

STEP	PERINTAH
1.	Start
2.	Buat cacahan data sama dengan 100d (64h) di register B
3.	Set alamat awal tempat simpan data 1900 di register IX
4.	Set pencatat cacahan register C = 0
5.	Ambil data dari memori yang alamatnya di catat oleh reg. IX simpan di register A
6.	Bandingkan data pada register A dengan 90
7.	Apakah Cy = 1
8.	Jika Ya lompat ke STEP 13
9.	Bandingkan data pada register A dengan 96
10.	Apakah Cy = 0
11.	Jika Ya lompat ke STEP 13
12.	Tambahkan nilai cacahan register C dengan 1
13.	Naikkan nilai pencatat alamat register IX satu alamat
14.	Kurangkan isi register B dengan 1
15.	Apakah nilai Z = 0
16.	Jika Ya kembali ke STEP 5
17.	Stop



Program :

ADDR	Kode Operasi	No.	Label	Assembly	Keterangan
1800	06 64	1		LD B , 64H	
1802	DD 21 00 19	2		LD IX , 1900H	
1806	0E 00	3		LD C , 00H	
1808	DD 7E 00	4	ULANG	LD A, (IX+00)	
180B	FE 90	5		CP, 90	
180D	DA 16 18	6		JP C, MAJU	
1810	FE 96			CP, 96	
1812	D2 16 18			JP NC, MAJU	
1815	0C	7		INC C	
1816	DD 23	8	MAJU	INC IX	
1818	10 EE	9		DJNZ, ULANG	
181A	FF	10		RST 38	

Pengujian Program :

Masukkan program di atas dan masukkan 100 data secara acak mulai alamat 1900 lalu eksekusi dan baca isi register C. Apakah isi register C menunjukkan jumlah cacahan data bernilai diantara 90 s/d 95 ?. Jika ya berarti program saudara sudah benar. Jika tidak cermati sekali lagi apakah masih ada kesalahan. Jangan lupa bertanya pada tutor saudara.

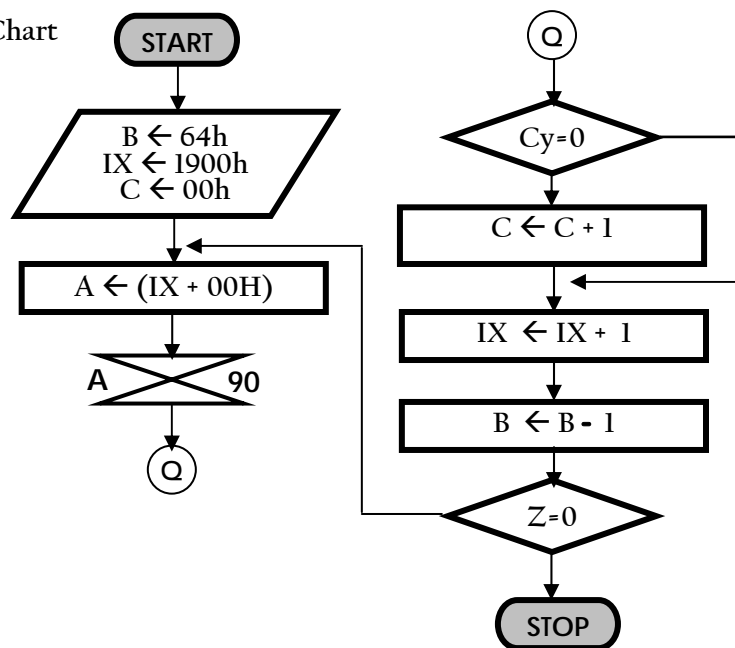
Penyelesaian Kasus f

Program untuk mencacah jumlah data bernilai lebih kecil dari 90 dari 100 data yang ada di memori mulai alamat 1900. Hasil cacahan dicatat di register C.

Algoritma:

STEP	PERINTAH
1.	Start
2.	Buat cacahan data sama dengan 100d (64h) di register B
3.	Set alamat awal tempat simpan data 1900 di register IX
4.	Set pencatat cacahan register C = 0
5.	Ambil data dari memori yang alamatnya di catat oleh reg. IX simpan di register A
6.	Bandingkan data pada register A dengan 90
7.	Apakah Cy = 0
8.	Jika Ya lompat ke STEP 10
9.	Tambahkan nilai cacahan register C dengan 1
10.	Naikkan nilai pencatat alamat register IX satu alamat
11.	Kurangkan isi register B dengan 1
12.	Apakah nilai Z = 0
13.	Jika Ya kembali ke STEP 5
14.	Stop

Flow Chart



Program :

ADDR	Kode Operasi	No.	Label	Assembly	Keterangan
1800	06 64	1		LD B , 64H	
1802	DD 21 00 19	2		LD IX , 1900H	
1806	0E 00	3		LD C , 00H	
1808	DD 7E 00	4	ULANG	LD A, (IX+00)	
180B	FE 90	5		CP, 90	
180D	D2 11 18	6		JP NC, MAJU	
1810	0C	7		INC C	
1811	DD 23	8	MAJU	INC IX	
1813	10 F3	9		DJNZ, ULANG	
1815	FF	10		RST 38	

Pengujian Program :

Masukkan program di atas dan masukkan 100 data secara acak mulai alamat 1900 lalu eksekusi dan baca isi register C. Apakah isi register C menunjukkan jumlah cacahan data bernilai lebih kecil dari 90 ?. Jika ya berarti program saudara sudah benar. Jika tidak cermati sekali lagi apakah masih ada kesalahan. Jangan lupa bertanya pada tutor saudara.

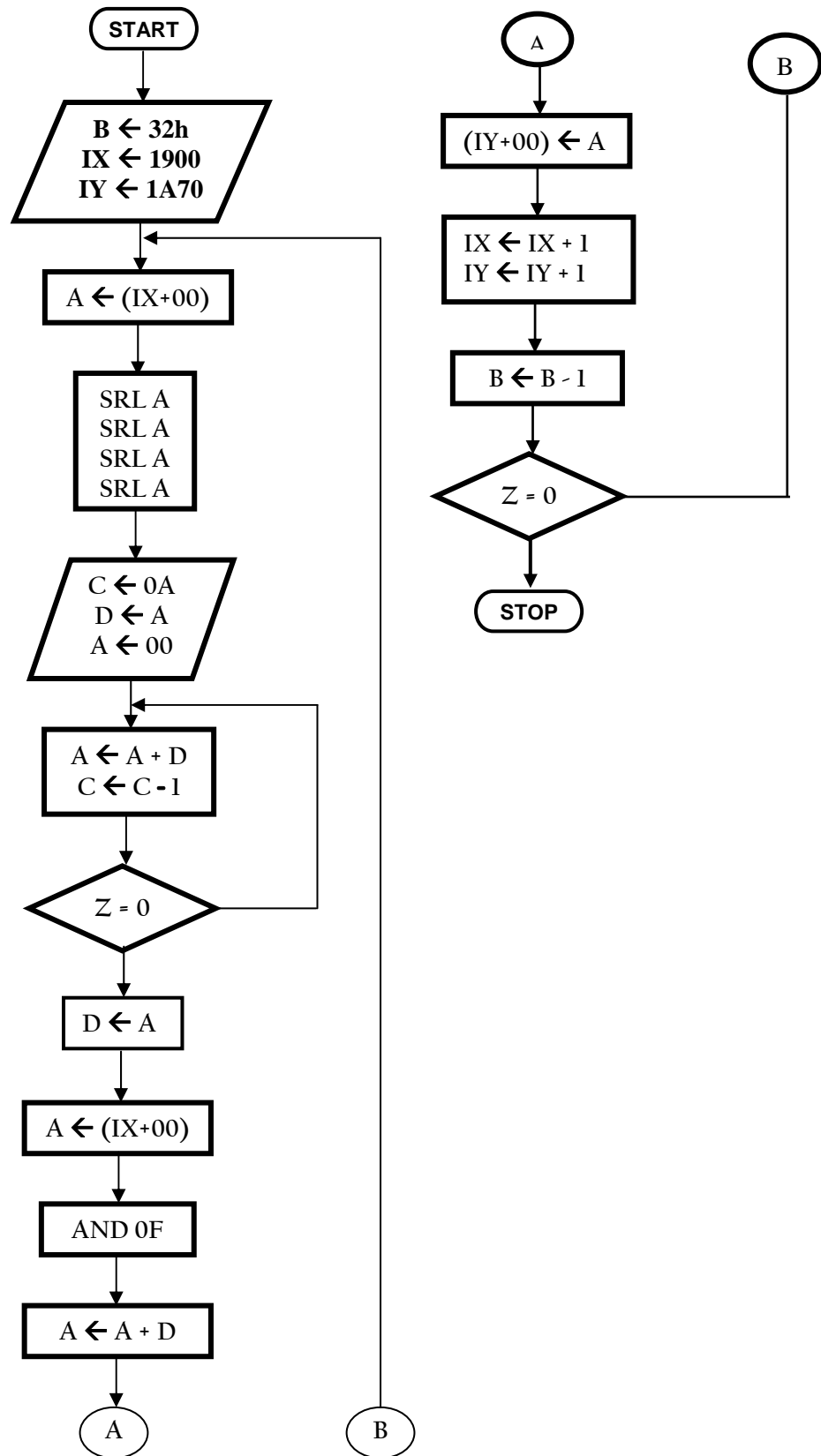
Penyelesaian Kasus g

Program untuk mengkonversi 50 bilangan Desimal menjadi bilangan Heksa/Biner.
Bilangan desimal tercatat di memori mulai alamat 1900 dan hasil konversi bilangan biner tercatat di memori muai alamat 1A00.

Algoritma:

STEP	PERINTAH
1.	Start
2.	Buat cacahan data sama dengan 50d (32h) di register B
3.	Set alamat awal tempat simpan data desimal 1900 di register IX
4.	Set alamat awal tempat simpan data biner 1A00 di register IY
5.	Ambil data dari memori yang alamatnya di catat oleh reg. IX simpan di register A
6.	Geser isi register A ke arah kanan sebanyak 4 kali
7.	Muat data 0A ke register C
8.	Muat data pada register A ke register D
9.	Muat data 00 ke register A
10.	Jumlahkan isi register D dengan isi register A
11.	Kurangi data pada register C dengan 1
12.	Apakah Z = 0, jika tidak kembali ke step 10
13.	Muat data pada register A ke register D
14.	Ambil data dari memori alamat tercatat register IX ke register A
15.	AND-kan isi register A dengan 0F
16.	Jumlahkan isi register D dengan isi register A
17.	Simpan data di register A ke memori tempat simpan data register IY
18.	Naikkan cacahan isi register IX dengan 1
19.	Naikkan cacahan isi register IY dengan 1
20.	Kurangi isi register B dengan 1
21.	Apakah Z = 0, jika tidak kembali ke STEP 5
22.	Stop

Flow Chart



Program :

ADD	Op Code	No	Label	Assembly	Keterangan
1800	06 32	1		LD B,32	
1802	DD 21 00 19	2		LD IX,1900	
1806	FD 21 00 1A	3		LD IY,1A00	
180A	DD 7E 00	4	awal	LD A,(IX+00)	
180D	CB 3F	5		SRL A	
180F	CB 3F	6		SRL A	
1811	CB 3F	7		SRL A	
1813	CB 3F	8		SRL A	
1815	0E 0A	9		LD C,0A	
1817	67	10		LD D,A	
1818	3E 00	11		LD A,00	
181A	82	12	LOOP	ADD A,D	
181B	0D	13		DEC C	
181C	C2 1A 18	14		JPNZ, LOOP	
181F	67	15		LD D,A	
1820	DD 7E 00	16		LD A,(IX+00)	
1823	E6 0F	17		AND 0F	
1825	82	18		ADD A,D	
1826	FD 77 00	19		LD (IY+00),A	
1829	DD 23	20		INC IX	
182B	FD 23	21		INC IY	
182D	5	22		DEC B	
182E	C2 0A 18	23		JPNZ,awal	
1831	FF	24		RST 38	

Pengujian Program :

Masukkan program di atas dan masukkan 50 data secara acak mulai alamat 1900 lalu eksekusi dan baca isi memori mulai alamat 1A00. Apakah isi data desimal pada alamat 1900 nilainya sama dengan isi data biner pada memori alamat 1A00. Demikian juga alamat berikutnya sampai alamat 1931. Jika tidak cermati sekali lagi apakah masih ada kesalahan. Jangan lupa bertanya pada tutor saudara.

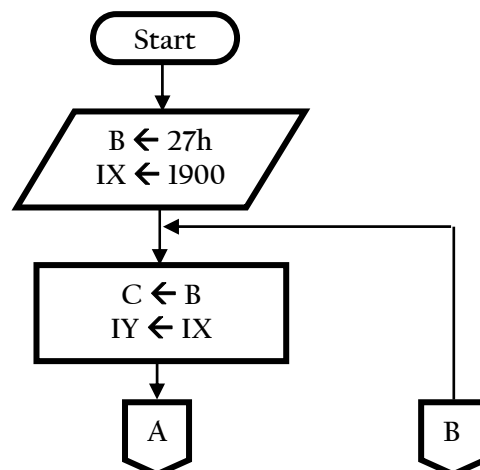
Penyelesaian Kasus h.

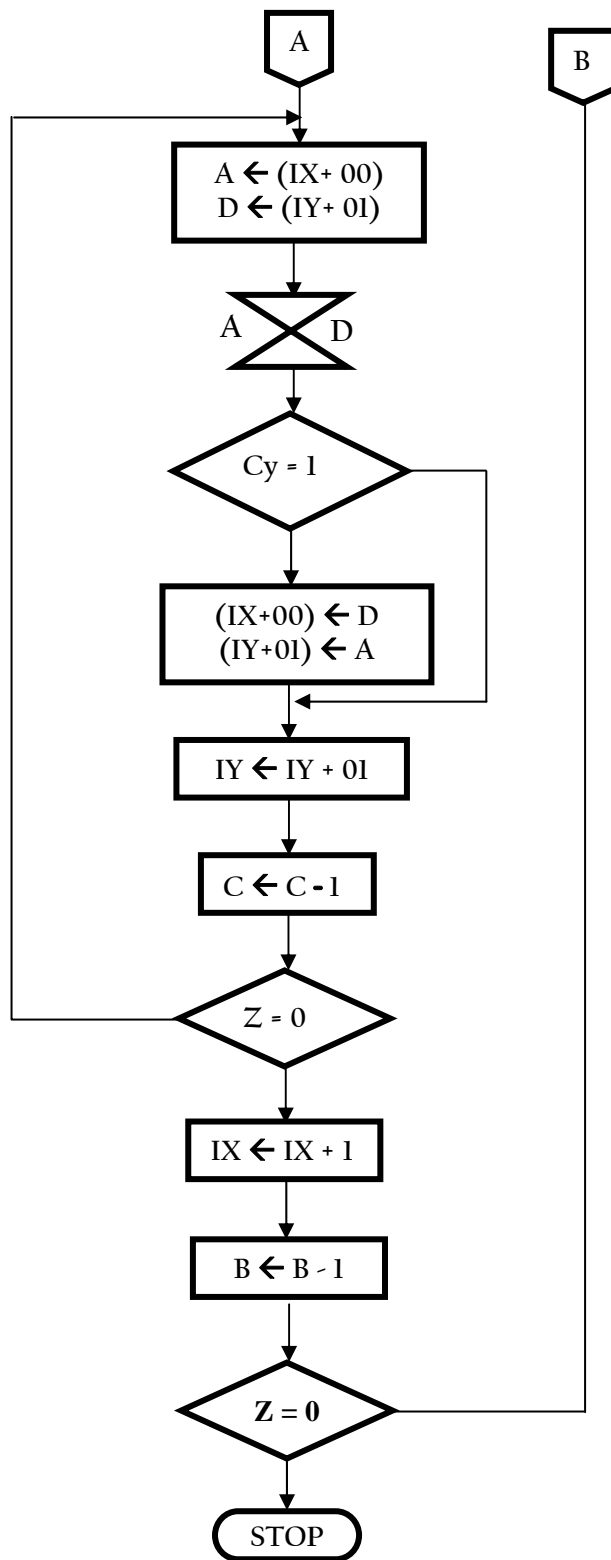
Program untuk mengurutkan 40 data dari kecil ke besar di memori mulai alamat 1900.

Algoritma:

STEP	PERINTAH
1.	Start
2.	Buat cacahan jumlah data = 27h di register B
3.	Set alamat awal tempat simpan data di 1900 di register IX
4.	Muat data pada register B ke register C
5.	Muat data pada register IX ke register IY
6.	Simpan data di memori tempat simpan data register IX ke register A
7.	Simpan data di memori tempat simpan data register IY +01 ke register D
8.	Bandingkan isi register A dengan isi register D
9.	Apakah Cy = 1, jika ya lompat ke step 12
10.	Muat data pada register D ke memori tempat simpan data register IX+00
11.	Muat data pada register A ke memori tempat simpan data register IY+01
12.	Naikkan isi register IY dengan 1
13.	Kurangi isi register C dengan 1
14.	Apakah Z = 0, jika tidak kembali ke step 6
15.	Naikkan isi register IX dengan 1
16.	Kurangi isi register B dengan 1
17.	Apakah Z = 0, jika tidak kembali ke step 4
18.	Stop

Flowchart:





Program :

Addr	Op Code	No	Label	Assembly	Keterangan
1800	06 27	1		LD B,27	
1802	DD 21 00 19	2		LD IX,1900	
1806	48	3	tiga	LD C,B	
1807	DD E5	4		PUSH IX	
1809	FD E1	5		POP IY	
180B	DD 7E 00	6	dua	LD A,(IX+00)	
180E	FD 56 01	7		LD D,(IY+01)	
1811	BA	8		CP D	
1812	DA IB 18	9		JPC, satu	
1815	DD 72 00	10		LD (IX+00), D	
1818	FD 77 01	11		LD (IY+01), A	
181B	FD 23	12	satu	INC IY	
181D	0D	13		DEC C	
181E	C2 0B 18	14		JPNZ, dua	
1821	DD 23	15		INC IX	
1823	5	16		DEC B	
1824	C2 06 18	17		JPNZ, tiga	
1827	FF	18		RST 38	

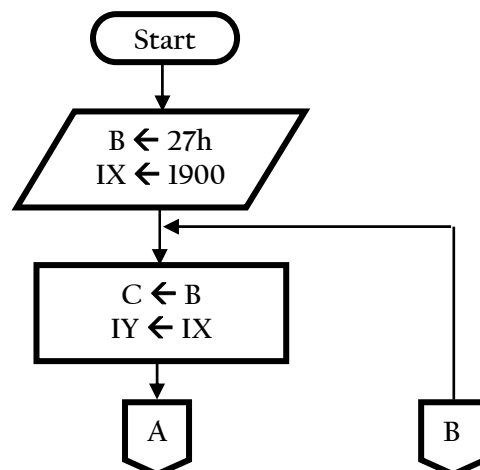
Penyelesaian Kasus I.

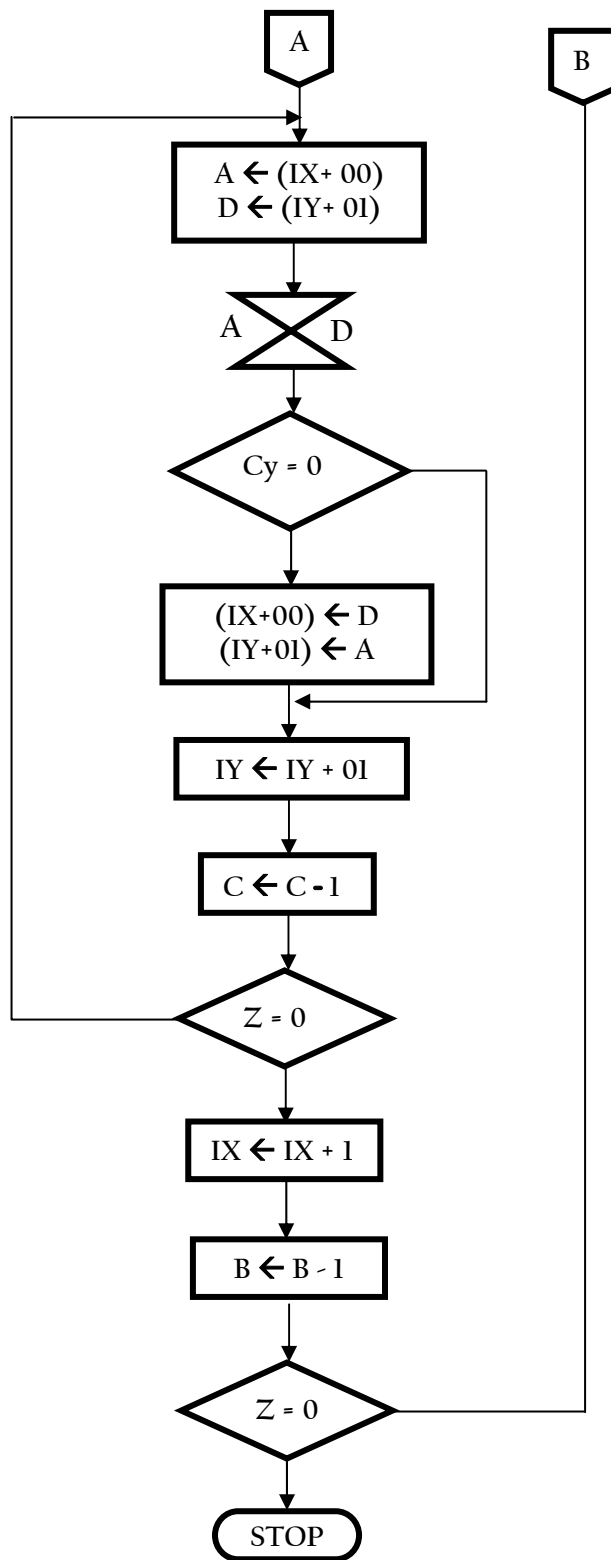
Program untuk mengurutkan 40 data dari besar ke kecil di memori mulai alamat 1900.

Algoritma:

STEP	PERINTAH
1.	Start
2.	Buat cacahan jumlah data = 27h di register B
3.	Set alamat awal tempat simpan data di 1900 di register IX
4.	Muat data pada register B ke register C
5.	Muat data pada register IX ke register IY
6.	Simpan data di memori tempat simpan data register IX ke register A
7.	Simpan data di memori tempat simpan data register IY +01 ke register D
8.	Bandingkan isi register A dengan isi register D
9.	Apakah Cy = 0, jika ya lompat ke step 12
10.	Muat data pada register D ke memori tempat simpan data register IX+00
11.	Muat data pada register A ke memori tempat simpan data register IY+01
12.	Naikkan isi register IY dengan 1
13.	Kurangi isi register C dengan 1
14.	Apakah Z = 0, jika tidak kembali ke step 6
15.	Naikkan isi register IX dengan 1
16.	Kurangi isi register B dengan 1
17.	Apakah Z = 0, jika tidak kembali ke step 4
18.	Stop

Flowchart:





Program :

Addr	Op Code	No	Label	Assembly	Keterangan
1800	06 27	1		LD B,27	
1802	DD 21 00 19	2		LD IX,1900	
1806	48	3	tiga	LD C,B	
1807	DD E5	4		PUSH IX	
1809	FD E1	5		POP IY	
180B	DD 7E 00	6	dua	LD A,(IX+00)	
180E	FD 56 01	7		LD D,(IY+01)	
1811	BA	8		CP D	
1812	D2 1B 18	9		JPNC, satu	
1815	DD 72 00	10		LD (IX+00), D	
1818	FD 77 01	11		LD (IY+01), A	
181B	FD 23	12	satu	INC IY	
181D	0D	13		DEC C	
181E	C2 0B 18	14		JPNZ, dua	
1821	DD 23	15		INC IX	
1823	5	16		DEC B	
1824	C2 06 18	17		JPNZ, tiga	
1827	FF	18		RST 38	

Lampiran

Contoh Program

DAFTAR PUSTAKA

- David Lalond, *The 8080, 8085, and Z80 Hardware, Software Programming, Interfacing, and Troubleshooting*, PHI, 1988
- Douglas VH., *Microprocessor and Interfacing Programming and Hardware*, MCGraw-Hill, 1992
- Hartono Partoharsodjo, *Dasar Pemrograman Mikroprosesor Zilog Z-80 di Mikrokomputer Micro-professor MPF-1*, FMIPA, ITB, Bandung 1982.
- Putu Sudira, *Kumpulan Lab Sheet Sistim Mikroprosesor*, Fakultas Teknik Universitas Negeri Yogyakarta, 2001
- S.H. Nasution, *Dari Chip ke Sistim : Pengantar Mikroprosesor*, Erlangga, Jakarta 1986
- Uffenbeck J., *The 8086/8088 Family Design, Programming and Interfacing*, PHI 1987