



BAB 2

Pemrograman Zilog Z80

Oleh :

Setiawardhana

Buku: Bahasa Assembly (Buku Komputer 3) oleh : Son Kuswadi

Politeknik Elektronika Negeri Surabaya



Pengantar

Question

- ◆ Mengapa sistem mikroprosesor yang sama dapat mengatur beragam sistem ?

Answer

- ◆ Adanya perangkat lunak, disamping perangkat keras penunjang lainnya



Pemrograman

- ◆ Definisi dan struktur
- ◆ Cara Assembly suatu program
- ◆ Flowchart program
- ◆ Listing Program



Kumpulan Instruksi

- ◆ Transfer Data
 - Antar Internal memori
 - Internal memori , register dan sebaliknya



Instruksi Input - Output

- ◆ Membaca dari port input
- ◆ Mengeluarkan data ke port output



Instruksi JUMP

- ◆ JUMP ke alamat yang dituju
- ◆ JUMP untuk inisialisasi
 - Inisialisasi register
 - Inisialisasi lokasi memori
 - Pemindahan data



Pemrograman

Definisi

- ◆ Kumpulan instruksi yang ditulis dengan format tertentu, bila dieksekusi oleh mikroprosesor maka mikroprosesor tersebut akan melaksanakan fungsi tertentu dengan hasil yang diperkirakan



Pemrograman

Struktur Instruksi Bahasa Assembly

- ◆ Operation Code (OpCode)
- ◆ Source Operand (Sumber)
- ◆ Destination Operand (Tujuan)

Contoh : LD A , H
 Opcode Source Destination

Instruksi : Pindahkan isi register 8 bit H ke
register 8 bit A, isi H = isi A



Opcode

- ◆ Contoh :

- LD
- LDI
- PUSH
- LDIR
- POP
- LPD
- CPIR
- INC
- CPD
- DAA
- ADD
- IM

- ◆ Mnemonic : Opcode yang ditulis dalam bentuk khusus

- ◆ Opcode biasanya disertai Operand

- Source Operand
- Destination Operand



Object Code

- ◆ Menulis program dalam assembly artinya menggunakan mnemonic
- ◆ Mnemonic untuk pembuat program bukan untuk mikroprosesor disebut Source Code
- ◆ Apa yang dibutuhkan mikroprosesor ?
- ◆ OBJECT CODE

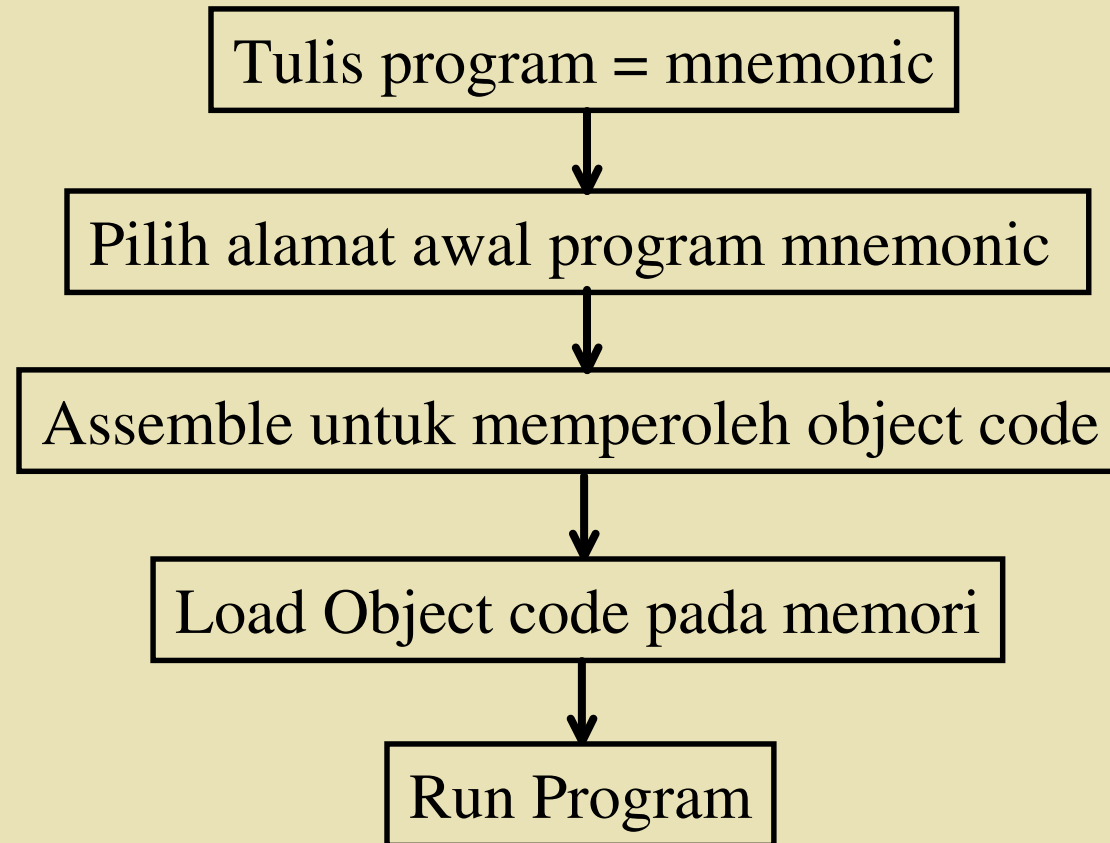


Object Code

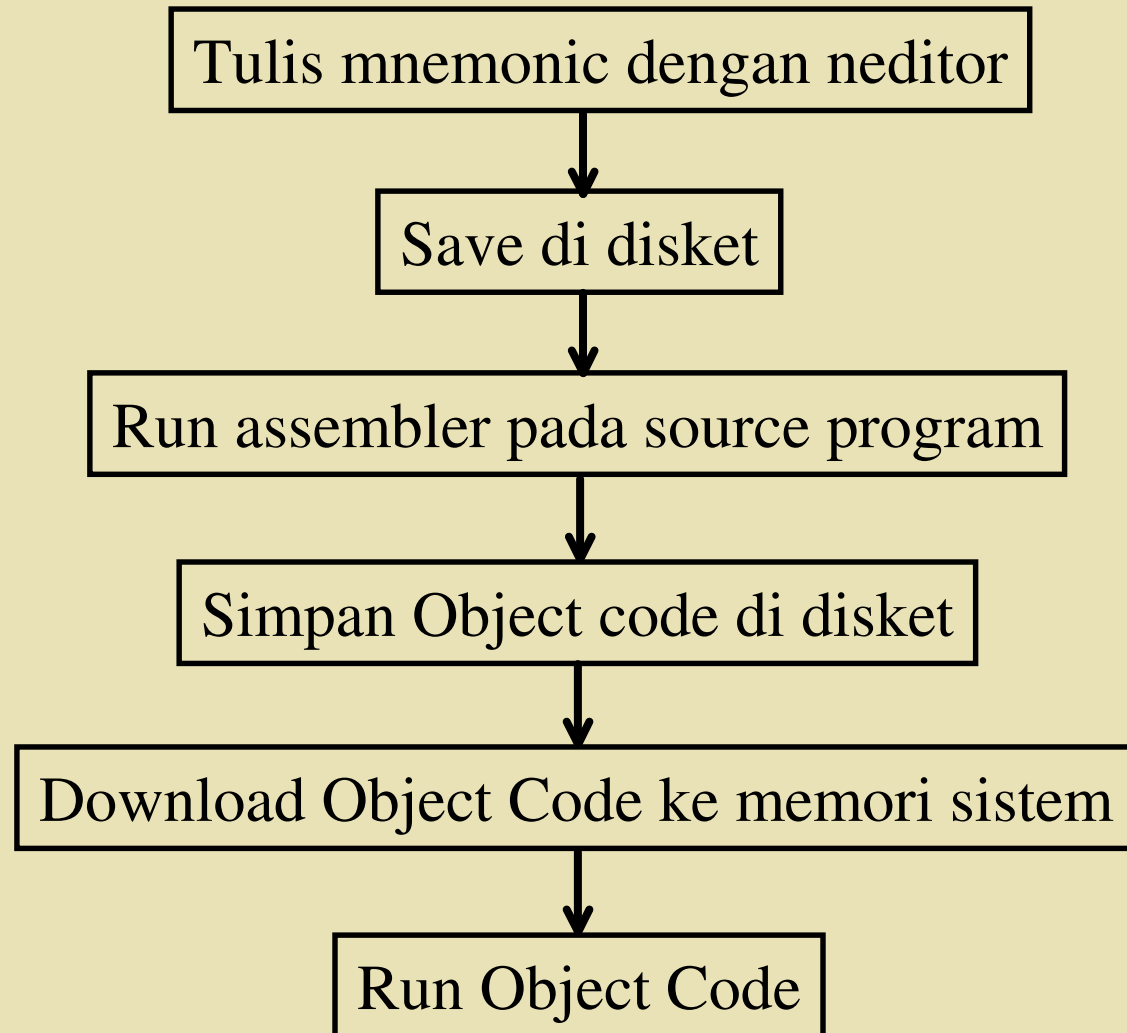
Source Code	Object Code	Biner
LD A,B	78H	0111 1000

Mikroprosesor membaca instruksi dalam bentuk biner di memori dan mengeksekusiannya

Cara Assembly Program



Cara Modifikasi Assemble





Flowchart

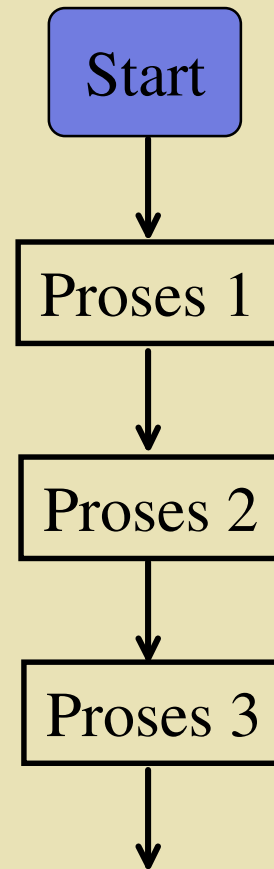
- ◆ Sistematis dalam pembuatan program
- ◆ Flowchart
 - Blok : Prosedur kasar
 - Rinci: Prosedur detail



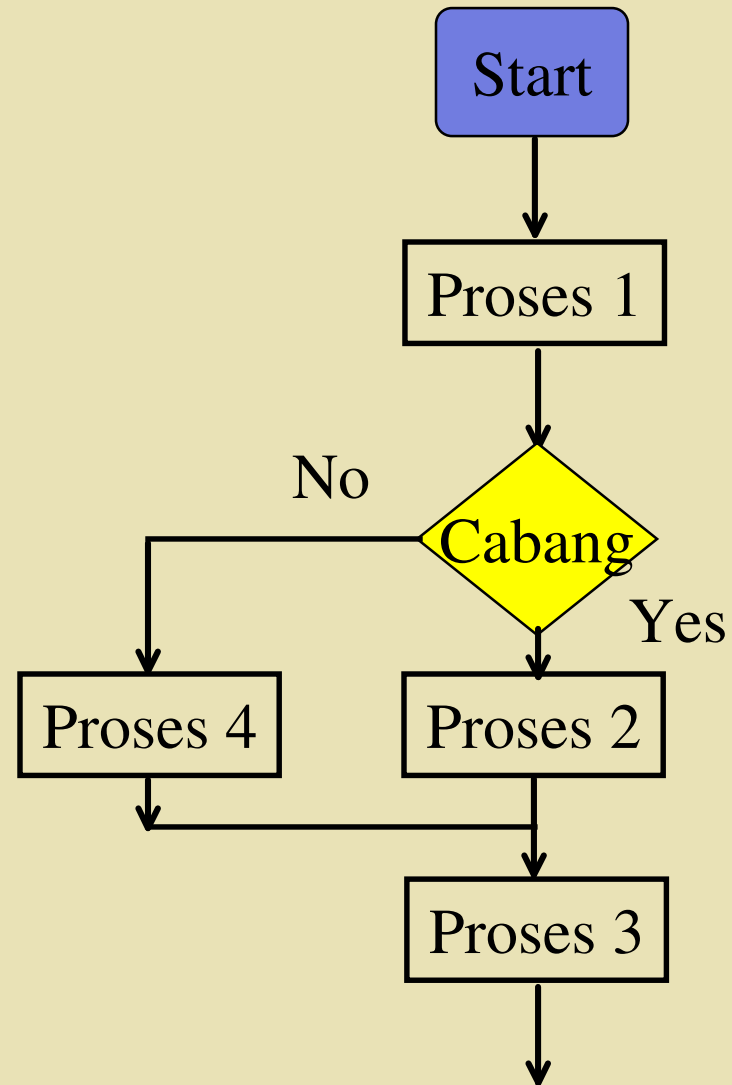
Flowchart

- ◆ Tipe langsung
- ◆ Tipe cabang
- ◆ Tipe loop

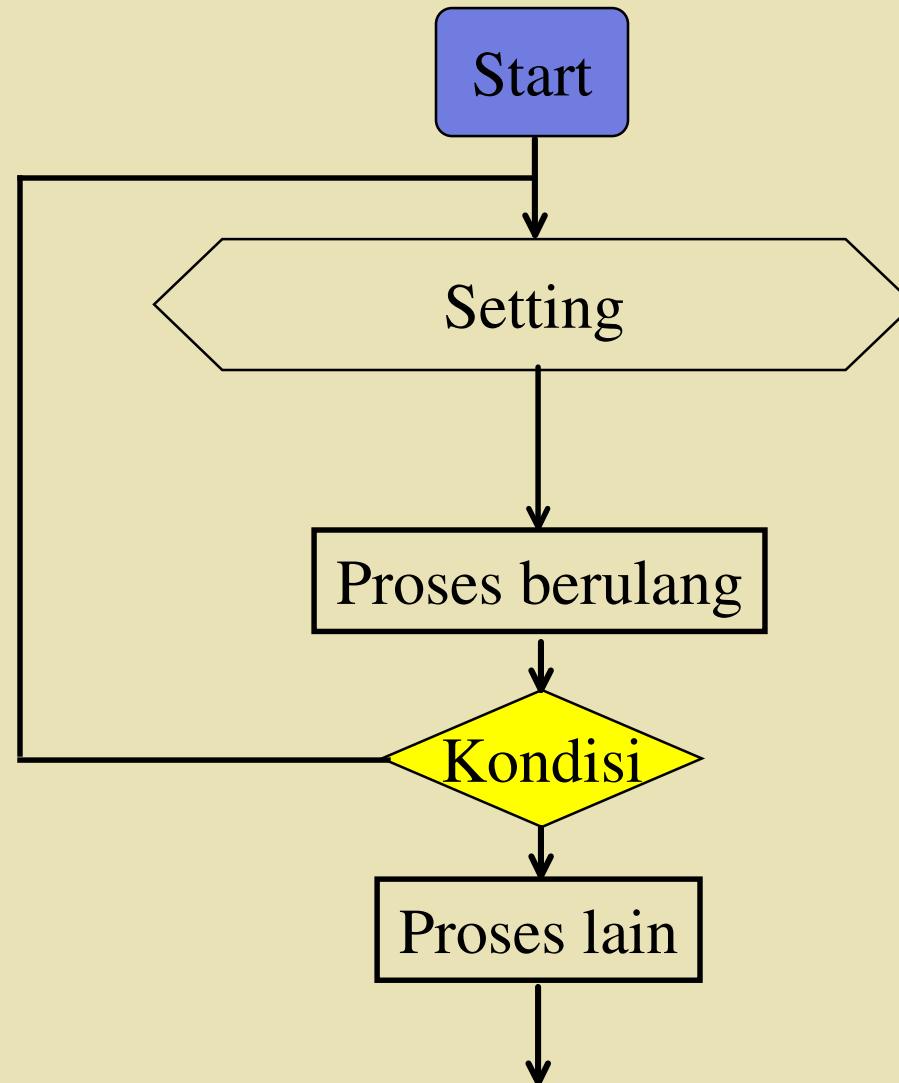
Type langsung (Straight Type)



Type Cabang (Branch Type)



Tipe Berulang (Loop Type)





Listing Program

- ◆ Object Code dihasilkan setelah listing program di assemble
- ◆ Selain Object Code juga dihasilkan listing program
- ◆ Listing program = Obj code + Source Code



Instruksi Z80

- ◆ Transfer Data
- ◆ Aritmatika
- ◆ Logika
- ◆ Branch
- ◆ Stac, I/O dan kontrol mesin



Transfer Data

- ◆ Internal register ke internal register (1)
- ◆ Memori sistem ke internal register (2)
- ◆ Contoh (1)

LD r1,r2

- ◆ Penjelasan
 - Instruksi mentransfer data dari source operand (r2) ke destination operand (r1).
 - Isi r2 tetap
 - Isi r1 berubah menjadi isi r2
 - Register yang dipakai : A,B,C,D,E,H dan L
- ◆ Object Code

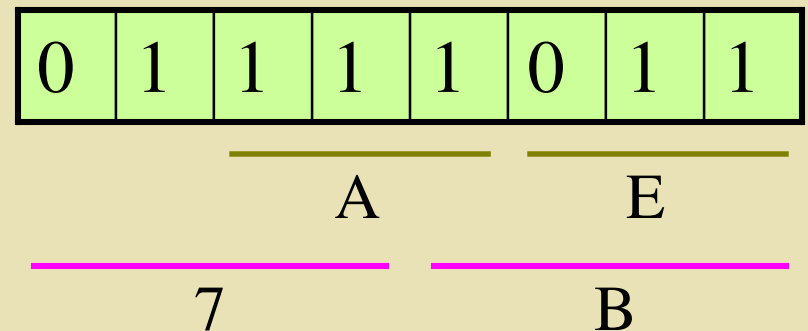
0	1	D	D	D	S	S	S
---	---	---	---	---	---	---	---

Kode register

- ◆ Register A = 111
- ◆ Register B = 000
- ◆ Register C = 001
- ◆ Register D = 010
- ◆ Register E = 011
- ◆ Register H = 100
- ◆ Register L = 101

Contoh : LD A, E

Object Code ? 7Bh =>





Transfer Data

- ◆ Contoh (2a)

LD A,(HL)

- ◆ Penjelasan

- Data di memori dipindah ke internal register
- Alamat dari memori : pasangan register H & L
- H : alamat orde tinggi (A15-A8)
- L : alamat orde rendah(A7-A0)

- ◆ Contoh :

- Alamat 8050h diisi 07h
- Pindahkan isinya ke register A
- Instruksi: LD HL,8050
LD A, (HL)



Transfer Data

- ◆ Contoh (2b)

LD (HL),r

- ◆ Penjelasan

- Data register dipindah ke memori sistem yang alamatnya ditunjuk oleh register H dan L

- ◆ Contoh

LD C,3Ah

LD HL,3E80h

LD (HL),C

Transfer Data

- ◆ Contoh (2c)

LD r,data

- ◆ Penjelasan

- Load immediate / reg. r dimuati segera dengan data 8 bit
- Operasi 2 byte
 - Byte pertama : Opcode
 - Byte kedua : 8 bit data

destination register byte pertama

0	0	D	D	D	1	1	0

data

byte kedua



Transfer Data

- ◆ Contoh (2d)

LD (HL),data

- ◆ Penjelasan

- Load immediate / alamat memori yang ditunjuk register HL akan dimuati segera dengan data 8 bit
- Operasi 2 byte
 - Byte pertama : Opcode
 - Byte kedua : 8 bit data

Contoh (alamat 80FFh diisi data 0FEh)

LD H,80

LD L,0FF

LD (HL),0FEh



Transfer Data

- ◆ Instruksi 3-byte

LD rp, data 16 bit

dimana :

- Byte 1 : Opcode
- Byte 2 dan 3 : data 16 bit

Transfer Data

- ◆ Instruksi 3-byte
- ◆ Kode pasangan register :
 - 00 =B,C
 - 01 =D,E
 - 10 =H,L
 - 11 =SP
- ◆ Detail operasinya :

pasangan register

0	0	R	P	0	0	0	1



Transfer Data

- ◆ Instruksi 3-byte
- ◆ Contoh LD (HL),54FEh
- ◆ Byte 1: 00 10 0001 = 0010 0001 = 21
- ◆ Byte 2: FEh
- ◆ Byte 3: 54h
- ◆ Object codenya : 21h FEh 54h



Transfer Data

- ◆ Instruksi 3-byte
- ◆ Contoh

LD (addr),A

- ◆ Penjelasan :
 - Menyimpan reg A ke memori dengan alamat address
 - Penyimpanan secara direct / langsung di memori



Transfer Data

- ◆ Contoh LD (8000h),A
- ◆ Object Code 32h 0Ah 80h

- ◆ Contoh LD A,(2436h)
- ◆ Object Code 3Ah 36h 24h



Transfer Data

- ◆ Contoh : 8050h diisi 7Ah 8051 diisi 80h
- ◆ Instruksi : LD HL,(8050h)
- ◆ Hasil :
 - Register L = 7Ah
 - Register H = 80h
 - Object code = 2Ah 50h 80h



Transfer Data

- ◆ Contoh : H diisi 40h dan L diisi 3Ah
- ◆ Instruksi : LD (80FEh),HL
- ◆ Hasil :
 - Alamat 80FEh berisi 3Ah
 - Alamat 80FFh berisi 40h



Transfer Data

◆ Contoh :

LD B,6Ah

LD C,7Dh

LD A,4Eh

LD (BC),A

◆ Hasil :

– Alamat 6A7Dh berisi 4Eh



Transfer Data

- ◆ Menukar data
- ◆ Instruksi : EX
- ◆ Contoh : EX DE,HL
- ◆ Hasil :
D \leftrightarrow H dan E \leftrightarrow L



Transfer Data

◆ Contoh :

LD HL,800Ah

LD DE,4326h

EX DE,HL

◆ Hasil :

– Semula : H=80h D=43h

L=0Ah E=26h

– Menjadi : H=43h D=80h

L=26h E=0Ah



Instruksi Input-Output

- ◆ INPUT > IN
- ◆ OUTPUT > OUT



Input

- ◆ Membaca dari port input
- ◆ Mnemonic IN A,(addr)
- ◆ Contoh IN A,(0FCh)
- ◆ Object Code DBh FCh



Output

- ◆ Mengirim data ke port output
- ◆ Mnemonic `OUT (addr),A`
- ◆ Contoh `OUT (0FDh),A`
- ◆ Object Code `D3h FEh`



Instruksi JUMP

- ◆ Instruksi “Lompat” ke “Alamat Tujuan”
- ◆ Instruksi : JP addr
- ◆ Contoh : JP 8043h
- ◆ Hasil : Lompat ke alamat 8043h
- ◆ Object Code : C3h 43h 80h







