

BAB 3

Pemrograman Zilog Z80 Lanjutan

Oleh :

Setiawardhana

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

Politeknik Elektronika Negeri Surabaya

Pokok Bahasan

- Addressing Mode ?
- FLAG ?
- Instruksi Aritmatika?
- Branch ?
- Instruksi Logika ?

Addressing mode

- Register
- Direct
- Immediate
- Register Indirect
- Indexed

Addressing mode - Register

- Perpindahan dari source operand internal register ke destination operand internal register
- Contoh :

LD A,B

LD B,C

Addressing mode - Direct

- Menyebut langsung alamat memori pada bagian instruksi
- Contoh : (Instruksi 3-byte)

LD A,(800Ah)

LD (8050h),A

Addressing mode - Immediate

- Data yang diolah merupakan bagian dari instruksi
- Contoh : (Instruksi 2 atau 3 byte)

LD A,4Eh

LD H,FFh

LD HL,8003h

Addressing mode – Register Indirect

- Menggunakan register internal (16 bit) atau pasangan register sebagai sumber dari alamat memori
- Contoh :

LD (HL),A

LD A,(DE)

LD (BC),A

Addressing mode - Indexed

- Register IX dan IY
- Bisa diisi data sembarang
- Sebagai penunjuk alamat source atau destination
- Contoh :

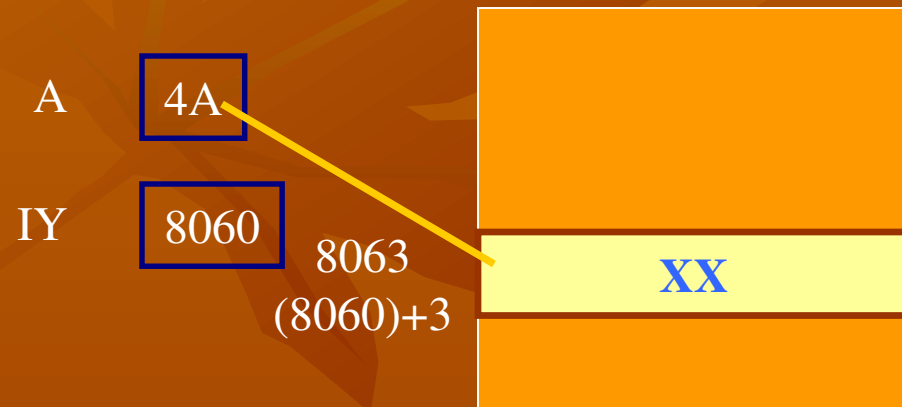
LD A,(IY+3)

LD (IX+2),B

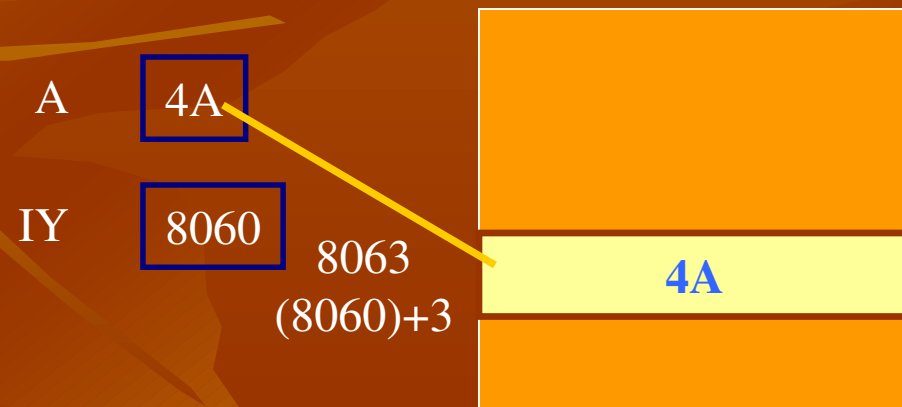
Addressing mode - Indexed

- Contoh :
- Register IP : 8060h
- Instruksi : LD ($IP+3$),A
- Hasil : memindahkan data dari register A ke alamat $8060h+3 = 8063h$

Konsep - Indexed Addressing



SEBELUM INSTRUKSI



SESUDAH INSTRUKSI

FLAG

- Menandai operasi CPU
- Flag register Z80 :

S	Z	X	H	X	P/V	N	CY
---	---	---	---	---	-----	---	----

CY

Carry Flag

P/V

Parity/Overflow Flag

H

Half Carry Flag

Z

Zero Flag

S

Sign Flag

N

Flag utk operasi desimal

Flag

- CY : Carry Flag
 - “1” = true , bila mikroprosesor membangkitkan carry internal ke carry bit
 - Operasi aritmatika dan logika
 - Bit ke-9 ?



Flag

- P/V

- Parity

- Contoh :

```
0101000
0001000
----- +
0110000 = parity genap
```

- Overflow

- Contoh : penambahan 2 bilangan positif

```
0111 1110 = +126
0000 0010 = +  2
----- +
1000 0000 = -128 (invalid = overflow)
```

- Contoh : penambahan 2 bilangan negatif

```
1000 0010 = -126
1000 1000 = -120
----- +
0000 1010 carry 1 (mestinya negatif, ttp overflow)
```

Flag

- Half Carry
- Flag “true” bila ada carry dari bit 3 dan 4
- Instruksi DAA (desimal adjust accumulation)
- DAA menangani 4 bit dan carry dari bit 3 ke bit 4

Flag

- Zero
- “true” bila hasil dari operasi adalah “nol”
- Contoh:

0111 1000

0111 1000 -

0000 0000

Flag

- Sign Flag
- “true” bila MSB dari hasil “1”
- Operasi aritmatika bertanda
- MSB sebagai penentu tanda bilangan

Flag

- N : flag
- Operasi BCD, 4 bit
- Instruksi DAA bergantung pada bit N
- N di set “1” setelah pengurangan
- N di set “0” setelah penambahans

Praktek Addressing mode

- Register

LD A,3Ah

LD B,A

LD C,B

LD D,C

LD E,D

HALT

Praktek Addressing mode

- Direct

```
LD A,(8040h)
```

```
OUT (0FEh),A
```

```
OUT (0FDh),A
```

```
HALT
```

Praktek Addressing mode

- Immediate / Register Indirect

LD A,36h

LD H,80h

LD L,40h

LD (HL),A

HALT

Praktek Addressing mode

- Indexed

```
LD IX,8040h
```

```
LD A,(IX+3)
```

```
OUT (0FDh),A
```

```
LD A,(IX+4)
```

```
OUT (0FEh),A
```

```
HALT
```

Instruksi Aritmatika

- Penambahan
- Pengurangan

Penambahan

- Isi register ke akumulator
ADD A,B
ADD A,C
- Isi memori ke akumulator
ADD A,(HL)
- Penambahan segera (immediate)
ADD A,25h

So'al :

Tambahkan angka 01F3h dengan 03A6h ?
Kurangkan angka 0100h dengan 00FFh ?

Penambahan - Penyelesaian

- Menambahkan dua data pada suatu alamat memori tertentu

```
ORG 8020H
LD HL,8040H
LD DE,8042H
LD BC,8044H
LD A,(HL)
EX DE,HL
ADD A,(HL)
LD (BC),A
INC HL
LD A,(HL)
EX DE,HL
INC HL
ADC A,(HL)
INC BC
LD (BC),A
HALT
```


Pengurangan - Penyelesaian

- Mengurangkan dua data pada suatu alamat memori tertentu

```
ORG 8050H
LD DE,8052H
LD BC,8054H
LD A,(HL)
EX DE,HL
SUB A,(HL)
LD (BC),A
EX DE,HL
INC HL
LD A,(HL)
EX DE,HL
INC HL
SBC A,(HL)
INC BC
LD (BC),A
HALT
```

Double Add pasangan register

- Menambah bilangan 16 bit dari pasangan register (BC,DE,HL,SP) ke pasangan register HL.
- Hasil disimpan di HL
- Mnemonic :

ADD HL,BC

ADD HL,DE

ADD HL,HL

ADD HL,SP

Increment

- 8 bit increment

INC r

INC (HL)

- 16 bit increment

INC BC

INC DE

INC HL

INC SP

Decrement

- 8 bit decrement

DEC r

DEC (HL)

- 16 bit decrement

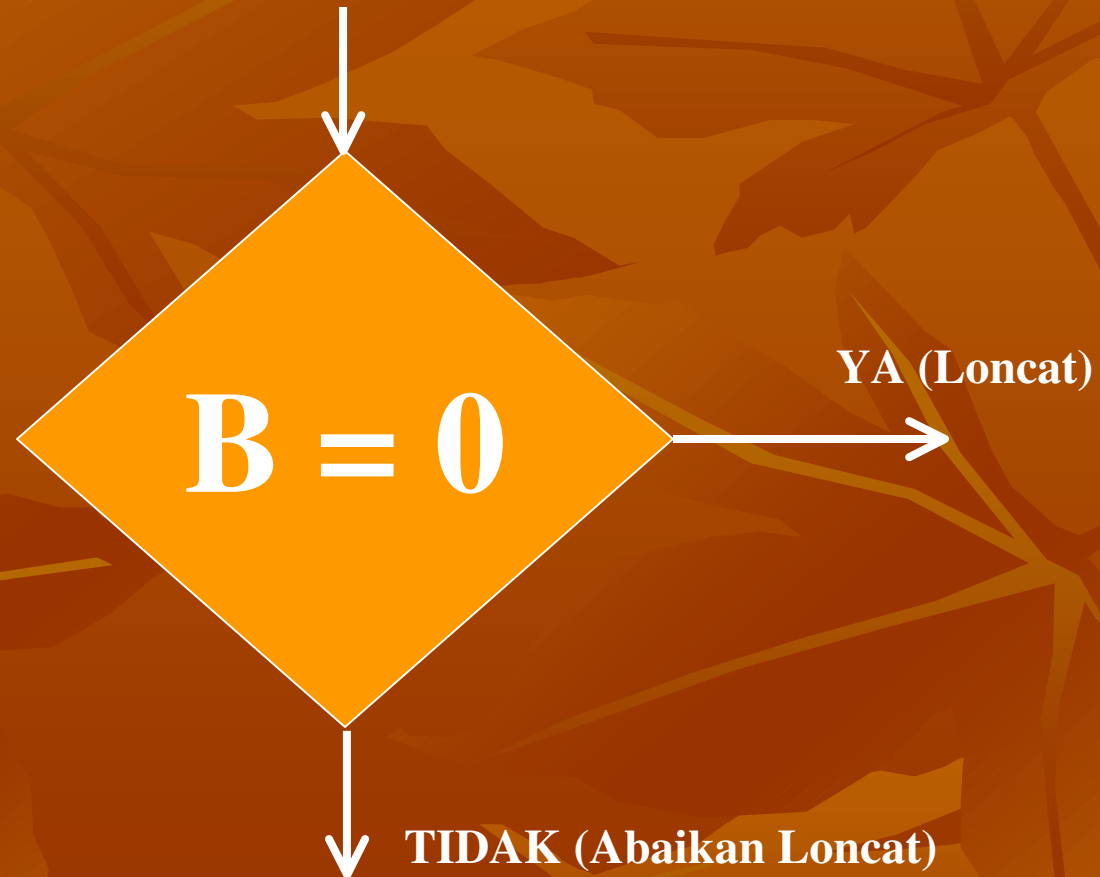
DEC BC

DEC DE

DEC HL

DEC SP

Percabangan bersyarat



Percabangan bersyarat

■ Mnemonic lengkap :

- JP Z,addr Jump bila Zero true
- JP NZ,addr Jump bila Zero false
- JP C,addr Jump bila Carry true
- JP NC,addr Jump bila Carry false
- JP PE,addr Jump bila Parity genap
- JP PO,addr Jump bila Parity ganjil
- JP M,addr Jump bila minus
- JP P,addr Jump bila plus

Instruksi Logika

- AND
- OR
- ExOR

AND

- Contoh

IN A,(0FCh)

AND 0BFh

A	0	1	1	0	0	1	0	1
B	1	1	0	0	0	1	1	1
Hasil	0	1	0	0	0	1	0	1

OR

- Contoh : IN A,(0FCh)
OR 0BFh

A	0	1	1	0	0	1	0	1
B	1	1	0	0	0	1	1	1
Hasil	1	1	1	0	0	1	1	1

XOR

- Contoh : IN A,(0FCh)
 XOR 0BFh

A	0	1	1	0	0	1	0	1
B	1	1	0	0	0	1	1	1
Hasil	1	0	1	0	0	0	1	0

Instruksi Compare

- Bandingkan 2 operand
- Set flag sesuai hasil compare
- Contoh

LD A,44h

LD B,45h

CP B

Proses : $44h - 45h = -1 = 0FFh$

Zero Flag = false

Carry Flag = true

Instruksi Rotate

- Rotate tanpa carry

RLCA

rotate kekiri

RRCA

rotate kekanan

Awal

1	0	1	0	0	1	0	1
0	1	0	0	1	0	1	1
1	1	0	1	0	0	1	0

RLCA

RRCA

Instruksi Rotate

- Rotate dengan carry

RLCA

rotate kekiri

RRCA

rotate kekanan

Data yang digeser disimpan ke register CY

Awal

1	0	1	0	0	1	0	1
0	1	0	0	1	0	1	0
0	1	0	1	0	0	1	0

RLCA

RRCA

Complement Accumulator, Carry dan Set Carry

- Complement Accumulator

Mnemonic : CPL

Bit CPL 1 jadi 0 dan 0 jadi 1

- Carry dan Set Carry

Mnemonic : CCF (Complement Carry Flag)

: SCF (Set Carry Flag)





