BAB 4
STACK AREA, SUBROUTINE dan INSTRUKSI BLOK

Oleh :
Setiawardhana

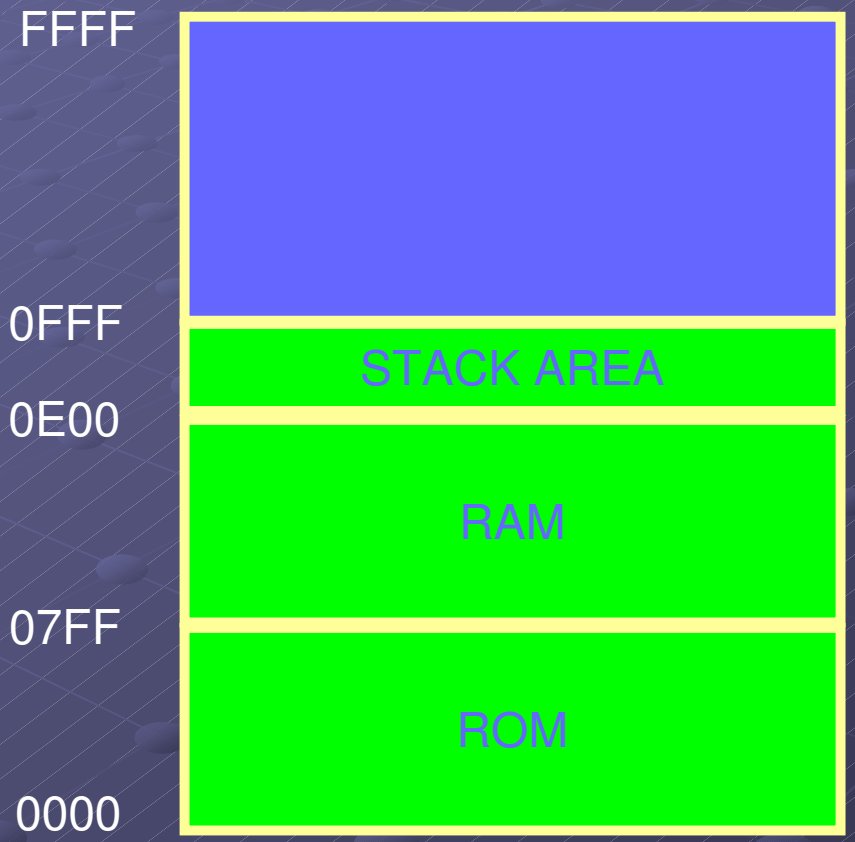
Buku: Bahasa Assembly (Buku Komputer 3) oleh : Son Kuswadi

Politeknik Elektronika Negeri Surabaya

STACK AREA

- Menyelamatkan register dalam operasi yang meloncat-loncat dari program utama ke program lain
- Pada daerah ini register dapat disimpan sementara dan dipanggil lagi dengan satu instruksi saja

STACK AREA

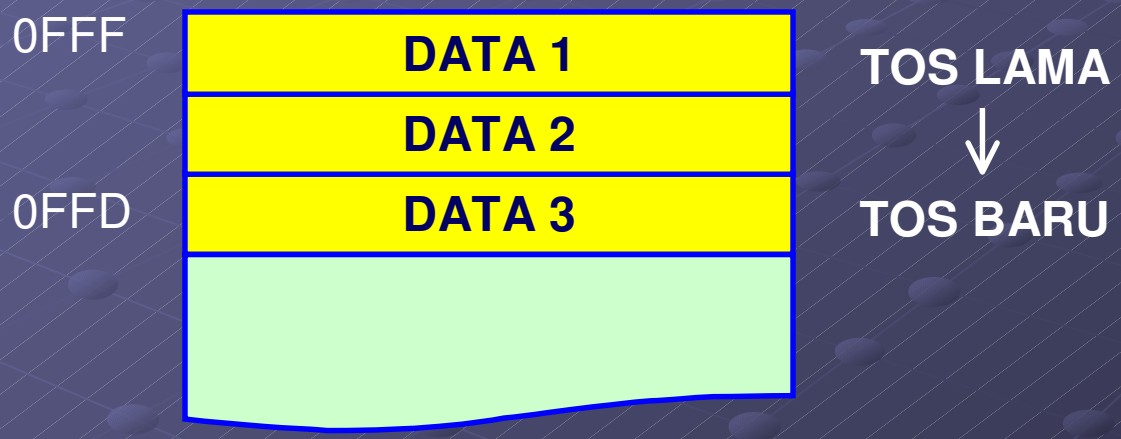


LOKASI STACK AREA

STACK AREA

- Stack Pointer = SP
- Internal register 16 bit khusus yang menjaga alamat dalam daerah ini
- SP selalu menunjuk TOS
- TOS = Top Of Stack

STACK AREA



Gambar TOS

STACK - TOS

- TOS harus di inisialisasi
- SP diset sembarang saat start up
- Instruksi set SP:

LD SP,data 16 bit

STACK

- Instruksi utama

PUSH	Menyimpan
POP	Mengambil

STACK - PUSH

- PUSH yang berlaku
  - PUSH BC
  - PUSH DE
  - PUSH HL
  - PUSH AF

STACK – Contoh Program PUSH

- Listing :

```
LD SP,3FFFh
LD   B,45h
LD   C,2Ah
PUSH BC
LD   D,89h
LD   E,1Eh
PUSH DE
```

STACK – Contoh Program PUSH

STACK – Flowchart PUSH

STACK – POP

- POP yang berlaku
  - POP BC
  - POP DE
  - POP HL
  - POP AF

STACK – Contoh Program POP

- Listing :

```
LD   BC,000h
LD   DE,000h
POP DE
POP BC
```

STACK – Contoh Program POP

STACK – Contoh Program POP

TOS setelah POP BC

TOS setelah POP DE

TOS awal

PROSES SESUDAH

SUBROUTINE

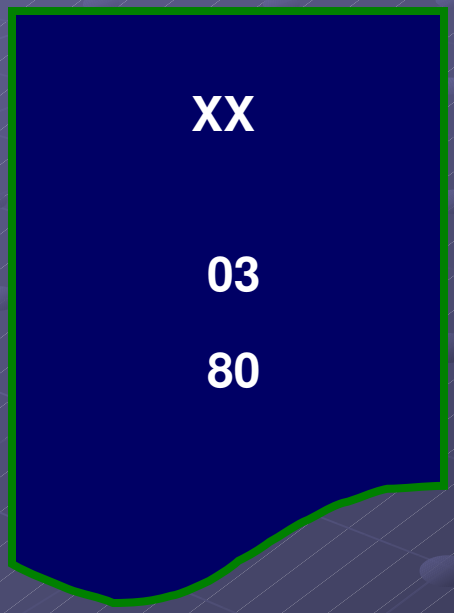
- Bila kita sering memanggil fungsi yang sama
- Instruksi : CALL subroutine
- Kembali ke program utama dengan :
  RET

SUBROUTINE

Object Code		Mnemonic
8000	C3	CALL TEST
8001	40	
8002	80	
8003		

Setelah CALL, PC berisi 8003, dan disimpan di SP

SUBROUTINE



TOS lama

TOS baru

SUBROUTINE

- Mnemonic CALL
  - CALL addr
  - CALL Z,addr
  - CALL NZ,addr
  - CALL C,addr
  - CALL NC,addr
  - CALL PE,addr
  - CALL PO,addr
  - CALL P,addr
  - CALL M,addr

SUBROUTINE

- Mnemonic RET
  - RET
  - RET Z
  - RET NZ
  - RET C
  - RET NC
  - RET PE
  - RET PO
  - RET P
  - RET M

Alternate Register

- Instruksi yang digunakan

  EX AF,AF’

  EXX

- Instruksi tersebut untuk menyelamatkan data secar cepat, bila terjadi interupsi, karena prosesnya lebih cepat dibanding PUSH

Index Register IX,IY

- Lebih mudah untuk mengakses sub kelompok data dari sekelompok data
- Contoh :

Alamat	Data blok
8500	| Nama sub blok 1
	| Umur
	| Tinggi
	| Berat
	| Telepon
8580	| Nama sub blok 2
	| Umur
	| Tinggi
	| Berat
	| Telepon

Index Register IX,IY

- Mnemonic yang digunakan :

```
LD   IX,8500H
LD   A,(IX+2)
```

Index Register IX,IY

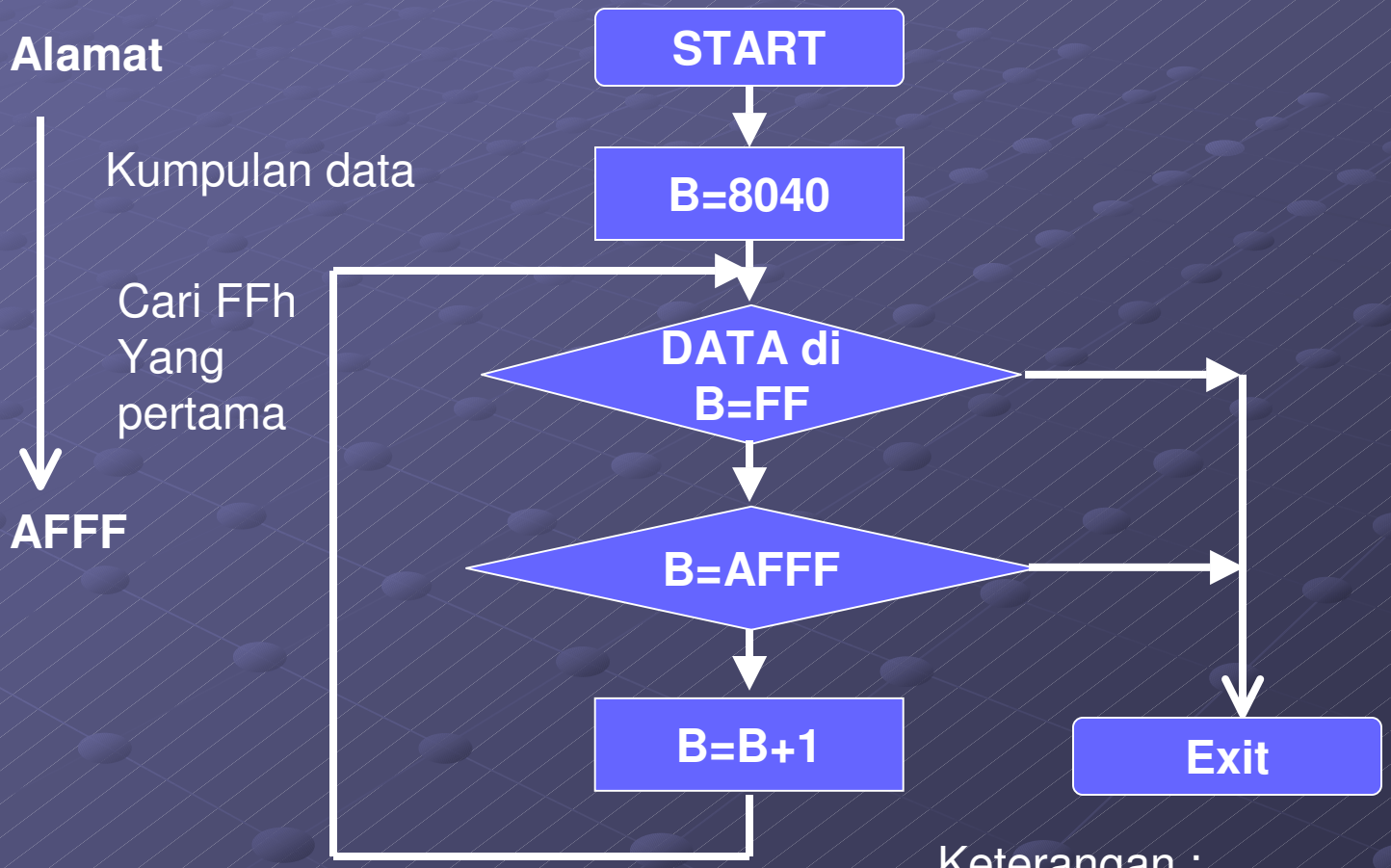
```
                8500        LD      DE,0080h
                8051        LD      IX,8500h
Subblok1        8052        LD      A,(IX+2)
                |           ADD IX,DE
                875F

                8580        LD      A,(IX+2)
                8581
Subblok1        8582
                |
                85FF
```

Instruksi Blok

- Instruksi yang bisa beroperasi dalam blok
- Bisa ditentukan batas alamat yang dicari
- Bisa memberikan informasi tentang ada atau tidaknya data

Instruksi Blok – Konsep Operasi



Keterangan :
Zero flag = True bila ketemu
Zero flag = False bila tidak ketemu

Instruksi Blok - Mnemonic

- LDI Load and Inc
- LDD Load and Dec
- LDIR Load Inc and Repeat
- LDDR Load Dec and Repeat
- CPI Compare and Inc
- CPD Compare and Dec
- CPIR Compare Inc and Repeat
- CPDR Compare Dec and Repeat

Instruksi Blok – Pasangan Register

- BC Untuk Counter 16 bit
- HL Memori pointer untuk source operand
- DE Memori pointer untuk destination operand

Instruksi LDI

- Memindahkan data 1 byte setiap kali dieksekusi.
- Data yang ditunjukkan HL(source) dipindah ke alamat yang ada pada DE (destination)
- Setelah instruksi maka:

HL = HL -1

DE = DE-1

BC = BC-1

Instruksi LDI

- Memindahkan data dari alamat 8050h ke alamat 8100h sebanyak 20 bytes

```
        LD    HL,8050h
        LD    DE,8100h
        LD    BC,20
LOOP:   LDI
        JP    PE,LOOP
        HALT
```

Instruksi - LDD

- Komsep Sama dengan LDI
- Perbedaannya :

HL = HL – 1
DE = DE – 1
BC = BC – 1

Instruksi - LDD

- Memindahkan data dari alamat 8000-83FF ke alamat 7400-77FF (1024 bytes data)

```
        LD   BC,1024
        LD   HL,83FFh
        LD   DE,77FFh
LOOP:   LDD
        JP   PE,LOOP
```

Instruksi LDIR dan LDDR

- Memindahkan 512 data dari alamat 8000-81FFFh ke alamat 9000-91FFFh

```
LD   BC,512
LD   HL,8000h
LD   DE,9000h
LDIR
```

Instruksi LDIR dan LDDR

- Memindahkan 512 data dari alamat 8000-81FFFh ke alamat 9000-91FFFh

```
LD   BC,512
LD   HL,81FFh
LD   DE,91FFh
LDDR
```

Instruksi LDIR dan LDDR

- Perbedaan LDIR dan LDDR
  - Waktu tanggapan terhadap Interupsi
  - Output dari Refresh Address

"Tidak menjadi bahasan sub materi disini"

Blok Compare Instruction

- Mnemonic
  - CPI Compare with Inc
  - CPD Compare with Dec
  - CPIR Compare Inc and Repeat
  - CPDR Compare Dec and Repeat

Pasangan register yang dipakai :

- HL Memori pointer yang di compare
- BC Byte counter

CPI dan CPD

- Isi dari data di memori yang alamatnya ditunjuk oleh register HL di compare dengan data di register A.
- Setelah di eksekusi :
  - Flag menunjukkan hasil compare
  - HL=HL$\pm$1
  - BC=BC-1

CPI

- Memindahkan data sebanyak 20 bytes dari alamat 8040h ke 8054h, bilangan yang di compare 55h

```
        LD    BC,20
        LD    HL,8040h
        LD    A,55h
LOOP:   CPI
        JP    Z,COCOK
        JP    PE,LOOP
```

CPD

- Memindahkan data sebanyak 20 bytes dari alamat 8040h ke 8054h, bilangan yang di compare 55h

```
        LD    BC,20
        LD    HL,8054h
        LD    A,55h
LOOP:   CPI
        JP    Z,COCOK
        JP    PE,LOOP
```

Instruksi CPIR dan CPDR

- Kemiripan operasi
  - CPIR Increment
  - CPDR Decrement
- Proses berhenti apabila
  - Register BC = 0
  - Register A cocok dengan data di memori (HL)

CPIR

```
LD   BC,20
LD   HL,8040h
LD   A,55h
CPIR
JP   Z,COCOK
```

CPDR

```
LD   BC,20
LD   HL,8054h
LD   A,55h
CPIR
JP   Z,COCOK
```