

MODE PENGALAMATAN

- Pada arsitektur 8086/88 memori = 1 MB artinya terdapat sejumlah $1 \text{ M} = 2^{20} = 1024$ sel memori 8 bit dengan alamat = 0 s/d $(2^{20} - 1) = 00000 \text{ H s/d FFFFF H}$
- Dalam arsitektur 8086/88 hanya terdapat register-register 16 bit yang hanya mampu mengamati memori dari 0 s/d $(2^{16}-1)$
- Oleh karena itu digunakan metode pengalamatan dengan dua register yaitu segment register dan offset register sehingga akan didapatkan pengalamatan 2^{20}
- Metode pengalamatan dengan dua register ini disebut dengan pengalamatan relatif (relative address)

Contoh :

Segment _____ _____ Offset

A123 : 099Ah

Komposisi Bit :

_____ _____
1010 0001 0010 0011 0000
0000 1001 1001 1010

1010 0001 1011 1100 1010 = A1BCA h

Alamat Mutlak (Physical Address) _____

- Ada beberapa cara (mode) untuk mengakses operand; Operand dapat terdapat di *register*, dalam *instruksi*, dalam *memori* atau *I/O*
- 8088 mempunyai sekitar 24 addressing mode → dikelompok menjadi 7:
 - Register Addressing
 - Immediate Addressing
 - Direct Addressing
 - Register Indirect Addressing
 - Base Relative Addressing
 - Direct Indexed Addressing
 - Base Indexed Addressing

ORGANISASI MEMORI

Pada arsitektur 8086/88 terdapat peta memori sbb :

Alamat Fisik	Keterangan
00000 – 0007F	BIOS Interrupt Vektor
00080 – 003FF	DOS Interrupt Vektor
00400 – 004FF	BIOS Data Area
00500 – 005FF	DOS & BASIC Data Area
00600 – 9CFFF	RAM Working Space
A0000 – A3FFF	Cadangan
A4000 – AFFFF	Monochrome Adapter
B0000 – B1000	Video Buffer
B1001 – B7FFF	Color/Graphics Adapter
B8000 – BBFFF	
BC000 – BFFFF	
C0000 – C7FFF	Ekspansi Memory
C8000 – C3FFF	Harddisk
CC000 – F3FFF	Cadangan
F4000 – F5FFF	User ROM (8 K)
F6000 – FDFFF	ROM BASIC (32 K)
FE000 - FFFFF	ROM BIOS (8 K)

1. Register dan Immediate Addressing

- Register Add.: Operand disimpan/diambil dari register
MOV AX,CX ; copy 16 bit isi CX (count regs) ke AX (acc. regs)
- Immediate Add.: source operand dapat berbentuk konstanta 8/16 bit; Konstanta ini terdapat dalam instruksi.
MOV CX,500
MOV CL,-30 ; simpan -30 ke reg. AL
K EQU 1024 ; source operand dapat berbentuk simbol yg didefine dgn EQU
- **INGAT:**
 - 8 bit signed number: 127(7H) s/d -128(80H)
 - 16 bit signed number: 32767(7FFFH) s/d -32768(8000H)
 - Max 8 bit unsigned #: 255 (0FFH)
 - Max 16 bit unsigned #: 65535 (0FFFFH)

2. Direct Addressing

MOV AX, TABLE ; load isi lokasi memori Table ke AX

3. Register Indirect Addressing

- Offset: effective address (EA) → jarak letak operand, dalam byte terhitung dari awal segment
- Address memory = offset + isi DS digeser dgn 0000
MOV AX,[BX] ; load isi memory yang ditunjuk BX ke AX

- Pertanyaan: bagaimana meletakkan EA ke BX?
 MOV BX,OFFSET TABLE
 MOV BX,OFFSET TABLE
 MOV AX,[BX]

4. Base Relative Addressing

- EA dihitung: Displacement + isi BX atau BP
 - BX cocok dipakai jika data yang hendak diakses terdapat di beberapa lokasi memori (terpisah)

MOV AX,[BX]+4 ;Load isi field record yang terdapat di byte ke 5 dan 6, ;dimana awal address record di BX ke AX

5. Direct Indexed Addressing

- EA = Displacement + Index Reg DI atau SI
 - DI atau SI cocok untuk akses elemen dalam Tabel
 - Displacement ke awal TABLE
 - Index reg: ke → elemen di dalam Tabel

MOV DI,4
 MOV AX,TABLE[DI] ;Load elemen ke3 ke AX TABLE adalah Tabel 1 Byte

6. Base Indexed Addressing

- Cocok untuk akses “two dimensional array”
- EA = Base regs + Index regs + Displ.
- Contoh:

MOV AX,[BX+2+DI] ;operand dapat ditulis
 MOV AX,[DI+BX+2] ;dlm [],urutan sembarang
 MOV AX,[BX+2][DI] ;displacement dpt digabung
 MOV AX,[BX][DI+2] ;dgn salah satu register