

# SET INSTRUKSI

Tipe-tipe Instruksi

1. Data Transfer (Register, memory, I/O Port)
2. Operasi Aritmatik
3. Manipulasi Bit (SHIFT, ROTATE, Logical dlm Register & Memory)
4. Control Transfer
5. String
6. Interrupt
7. Process Control

## 1. DATA TRANSFER

Berfungsi memindahkan data dan address diantara register dan lokasi memori atau I/O Port

Terbagi menjadi :

- General Purpose : MOV, PUSH, POP, XCHG, XLAT
- I/O : IN, OUT
- Address Transfer : LEA, LDS, LES
- Flag Register : LAHF, SAHF, PUSHF, POPF

### General Purpose

#### 1.1 MOV (MOVE)

Perintah untuk mengisi, meindahkan, memperbaharui isi suatu register, variabel ataupun suatu lokasi memory

MOV [operand1], [operand2]

Contoh :  
MOV AX, TABLE  
MOV TABLE, AX  
MOV ES:[BX], AX  
MOV DS, AX  
MOV BL, AL  
MOV CL, -30  
MOV DEST, 25h

## 1.2 XCHG (EXCHANGE)

Untuk mempertukarkan isi source operand dengan destination operand dalam ukuran byte atau word

Dapat mempertukarkan isi 2 memori dan isi suatu register dan lokasi memori

Isi segmen register tdk dapat dipertukarkan (dengan instruksi XCHG)

```
XCHG AX,BX           ;pertukaran 2 word register
XCHG AL,BH           ;pertukaran 2 byte register
XCHG WORD_LOC,DX    ;pertukaran dengan lokasi memory
XCHG DL,BYTE_LOC    ;pertukaran dengan register
```

## 1.3 XLAT (TRANSLATE)

Untuk membaca/memeriksa nilai-nilai (byte) yang terdapat dalam suatu TABEL dan kemudian megkopi (load) ke dalam register AL (ukuran max TABEL=256 byte)

```
XLAT source_table
```

Sebelum eksekusi XLAT → starting address daripada tabelnya harus disimpan (load) ke BX dan index dari byte yg dimaksud ke dalam AL  
XLAT cocok untuk konversi yang membutuhkan waktu perhitungan yang lama misal konversi ASCII display code dari hexadesimal digit.

Contoh : Urutan instruksi utk memeriksa byte ke-10 dari tabel S\_TAB :

```
MOV AL,10             ;load harga index AL
MOV BX,OFFSETS_TAB   ;load starting add S_TAB ke BX
XLAT S_TAB            ;ambil harga dalam tabel ke AL
```

## **Address Transfer**

Transfer address suatu variabel → load address ke register mirip dengan instruksi move

### 2.1 LEA (Load Effective Address)

Misal DI=5

```
LEA BX,TABLE[DI]     ;load offset address dari TABLE+5 ke BX
```

## 2.2 LDS (Load Pointer using DS)

Membaca 32 bit dari memori kemudian load 16 bit pertama ke register tertentu dan 16 bit ke dua ke DS

LDS register16,memory32

## 2.3 LES(Load Pointer using ES)