



ASSALAMU'ALAIKUM
ARSITEKTUR KOMPUTER

SET INSTRUKSI

RAHMAD KURNIAWAN, S.T., M.I.T.

TEKNIK INFORMATIKA UIN SUSKA RIAU

Karakteristik dan Fungsi Set Instruksi

- ❖ Operasi dari CPU ditentukan oleh instruksi-instruksi yang dilaksanakan atau dijalankannya. Instruksi ini sering disebut sebagai instruksi mesin (*mechine instructions*) atau instruksi komputer (*computer instructions*).
- ❖ Kumpulan dari instruksi-instruksi yang berbeda yang dapat dijalankan oleh CPU disebut set Instruksi (*Instruction Set*).

Elemen-elemen Instruksi Mesin (Set Instruksi)

- ***Operation Code (opcode)*** : menentukan operasi yang akan dilaksanakan
- ***Source Operand Reference*** : merupakan input bagi operasi yang akan dilaksanakan
- ***Result Operand Reference*** : merupakan hasil dari operasi yang dilaksanakan
- ***Next instruction Reference*** : memberitahu CPU untuk mengambil (*fetch*) instruksi berikutnya setelah instruksi yang dijalankan selesai.

❖ ***Source*** dan ***result operands*** dapat berupa salah satu diantara tiga jenis berikut ini:

- *Main or Virtual Memory*
- *CPU Register*
- *I/O Device*

❖ **Desain set instruksi merupakan masalah yang sangat kompleks yang melibatkan banyak aspek, diantaranya adalah:**

- Kelengkapan set instruksi
- Ortogonalitas (sifat independensi instruksi)
- Kompatibilitas :
 - *Source code compatibility*
 - *Object code Compatibility*

❖ Selain ketiga aspek tersebut juga melibatkan hal-hal sebagai berikut:

- Operation Repertoire: Berapa banyak dan operasi apa saja yang disediakan, dan berapa sulit operasinya
- Data Types: tipe/jenis data yang dapat olah
- Instruction Format: panjangnya, banyaknya alamat, dsb.
- Register: Banyaknya register yang dapat digunakan
- Addressing: Mode pengalamatan untuk operand

Format Instruksi

- ❖ Suatu instruksi terdiri dari beberapa *field* yang sesuai dengan elemen dalam instruksi tersebut. Layout dari suatu instruksi sering disebut sebagai Format Instruksi (*Instruction Format*).

OPCODE	OPERAND REFERENCE	OPERAND REFERENCE
--------	----------------------	----------------------

- ❖ **Addresses (akan dibahas pada addressing modes)**
- ❖ **Numbers :**
 - Integer or fixed point
 - Floating point
 - Decimal (BCD)
- ❖ **Characters :**
 - ASCII
 - EBCDIC
- ❖ **Logical Data : Bila data berbentuk binary: 0 dan 1**

- 1. Data processing: *Arithmetic dan Logic Instructions***
- 2. Data storage: *Memory instructions***
- 3. Data Movement: *I/O instructions***
- 4. Control: *Test and branch instructions***

- ❖ **Menetapkan lokasi operand sumber dan operand tujuan.**
- ❖ **Lokasi-lokasi tersebut dapat berupa memori, register atau bagian paling atas daripada stack.**
- ❖ **Menetapkan panjang data yang dipindahkan.**
- ❖ **Menetapkan mode pengalamatan.**
- ❖ **Tindakan CPU untuk melakukan transfer data adalah :**
 - Memindahkan data dari satu lokasi ke lokasi lain.
 - Apabila memori dilibatkan :
 - Menetapkan alamat memori.
 - Menjalankan transformasi alamat memori virtual ke alamat memori aktual.
 - Mengawali pembacaan / penulisan memori

Operasi set instruksi untuk transfer data :

- ❖ **MOVE** : memindahkan word atau blok dari sumber ke tujuan
- ❖ **STORE** : memindahkan word dari prosesor ke memori.
- ❖ **LOAD** : memindahkan word dari memori ke prosesor.
- ❖ **EXCHANGE** : menukar isi sumber ke tujuan.
- ❖ **CLEAR / RESET** : memindahkan word 0 ke tujuan.
- ❖ **SET** : memindahkan word 1 ke tujuan.
- ❖ **PUSH** : memindahkan word dari sumber ke bagian paling atas stack.
- ❖ **POP** : memindahkan word dari bagian paling atas sumber

- ❖ **Tindakan CPU untuk melakukan operasi arithmetic :**
 1. Transfer data sebelum atau sesudah.
 2. Melakukan fungsi dalam ALU.
 3. Menyet kode-kode kondisi dan flag.
- ❖ **Operasi set instruksi untuk arithmetic :**

1. ADD : penjumlahan	5. ABSOLUTE
2. SUBTRACT : pengurangan	6. NEGATIVE
3. MULTIPLY : perkalian	7. DECREMENT
4. DIVIDE : pembagian	8. INCREMENT

Nomor 5 sampai 8 merupakan instruksi operand tunggal.

- ❖ **Tindakan CPU sama dengan arithmetic**
- ❖ **Operasi set instruksi untuk operasi logical :**
 - AND, OR, NOT, EXOR
 - COMPARE : melakukan perbandingan logika.
 - TEST : menguji kondisi tertentu.
 - SHIFT : operand menggeser ke kiri atau kanan menyebabkan konstanta pada ujung bit.
 - ROTATE : operand menggeser ke kiri atau ke kanan dengan ujung yang terjalin.

- ❖ **Tindakan CPU sama dengan arithmetic dan logical.**
- ❖ **Instruksi yang mengubah format instruksi yang beroperasi terhadap format data.**
- ❖ **Misalnya pengubahan bilangan desimal menjadi bilangan biner.**
- ❖ **Operasi set instruksi untuk konversi :**
 - **TRANSLATE** : menterjemahkan nilai-nilai dalam suatu bagian memori berdasarkan tabel korespondensi.
 - **CONVERT** : mengkonversi isi suatu word dari suatu bentuk ke bentuk lainnya.

❖ **Tindakan CPU untuk melakukan INPUT / OUTPUT :**

- Apabila memory mapped I/O maka menentukan alamat memory mapped.
- Mengawali perintah ke modul I/O

❖ **Operasi set instruksi Input / Output :**

- INPUT : memindahkan data dari perangkat I/O tertentu ke tujuan
- OUTPUT : memindahkan data dari sumber tertentu ke perangkat I/O
- START I/O : memindahkan instruksi ke prosesor I/O untuk mengawali operasi I/O
- TEST I/O : memindahkan informasi dari sistem I/O ke tujuan

- ❖ **Tindakan CPU untuk transfer control: Mengupdate program counter untuk subrutin, call / return.**
- ❖ **Operasi set instruksi untuk transfer control:**
 - JUMP (cabang) : pemindahan tidak bersyarat dan memuat PC dengan alamat tertentu.
 - JUMP BERSYARAT : menguji persyaratan tertentu dan memuat PC dengan alamat tertentu atau tidak melakukan apa tergantung dari persyaratan.
 - JUMP SUBRUTIN : melompat ke alamat tertentu.
 - RETURN : mengganti isi PC dan register lainnya yang berasal dari lokasi tertentu.
 - EXECUTE : mengambil operand dari lokasi tertentu dan mengeksekusi sebagai instruksi

- SKIP : menambah PC sehingga melompati instruksi berikutnya.
- SKIP BERSYARAT : melompat atau tidak melakukan apa-apa berdasarkan pada persyaratan
- HALT : menghentikan eksekusi program.
- WAIT (HOLD) : melanjutkan eksekusi pada saat persyaratan dipenuhi.
- NO OPERATION : tidak ada operasi yang dilakukan.

- ❖ **Hanya dapat dieksekusi ketika prosesor berada dalam keadaan khusus tertentu atau sedang mengeksekusi suatu program yang berada dalam area khusus, biasanya digunakan dalam sistem operasi.**
- ❖ **Contoh : membaca atau mengubah register kontrol.**

Jumlah Alamat (Number Of Addresses)

- ❖ Salah satu cara tradisional untuk menggambarkan arsitektur prosesor adalah dengan melihat jumlah alamat yang terkandung dalam setiap instruksinya.
- ❖ Jumlah alamat maksimum yang mungkin diperlukan dalam sebuah instruksi :
 - Empat Alamat (dua operand, satu hasil, satu untuk alamat instruksi berikutnya)
 - Tiga Alamat (dua operand, satu hasil)
 - Dua Alamat (satu operand merangkap hasil, satunya lagi operand)
 - Satu Alamat (menggunakan accumulator untuk menyimpan operand dan hasilnya)

Macam-macam Instruksi Menurut Jumlah Operasi Yang Dispesifikasikan

- 1. O – Address Instruction**
- 2. 1 – Address Instruction.**
- 3. N – Address Instruction**
- 4. M + N – Address Instruction**

Macam-macam instruksi menurut sifat akses terhadap memori atau register

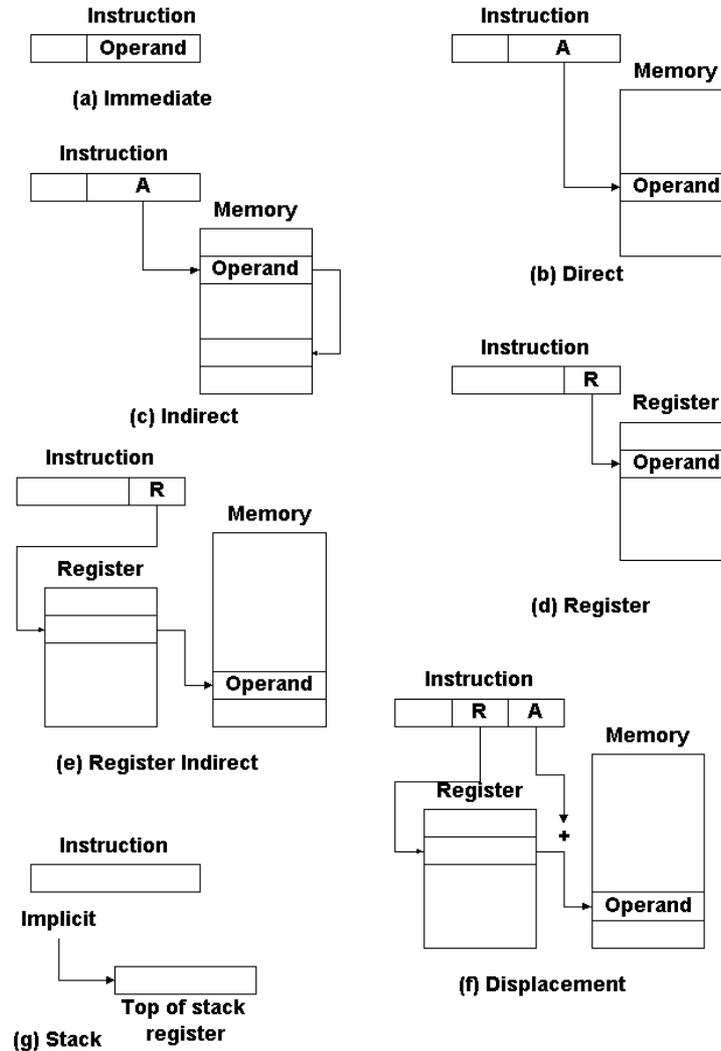
- 1. Memori To Register Instruction**
- 2. Memori To Memori Instruction**
- 3. Register To Register Instruction**

Jenis-jenis addressing modes (Teknik Pengalamatan) yang paling umum:

- ❖ *Immediate*
- ❖ *Direct*
- ❖ *Indirect*
- ❖ *Register*
- ❖ *Register Indirect*
- ❖ *Displacement*
- ❖ *Stack*

Mode	Algorithm	Principal Advantage	Principal Disadvantage
Immediate	Operand = A	No memory reference	Limited operand magnitude
Direct	EA = A	Simple	Limited address space
Indirect	EA = (A)	Large address space	Multiple memory references
Register	EA = R	No memory Reference	Limited address space
Register Indirect	EA = (R)	Large address space	Extra memory reference
Displacement	EA=A+(R)	flexibility	Complexity
Stack	EA=top of Stack	No memory Reference	Limited applicability

Gambar Addressing Mode





❖ **Thank you**