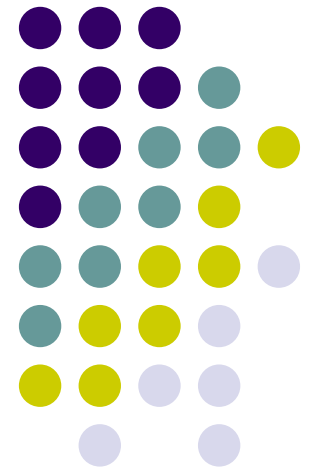
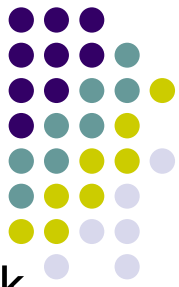


BAHASA TINGKAT RENDAH



PEMROGRAMAN BAHASA MESIN

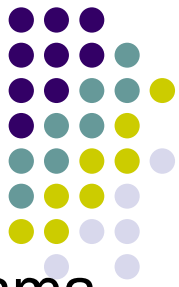


- instruksi mesin akan ditunjukkan dalam oktal dan juga biner utk menunjukkan bagaimana pengkodean oktal membantu programmer.
- Dalam penulisan MNEMONICS notasi alamat akan ditunjukkan sebagai berikut :
 1. STA 7 : simpan isi akumulator dalam lokasi 7
 2. STA@7 : simpan isi akumulator dalam lokasi 7(@ menunjukkan alamat tak langsung)
 3. STA 5,X : tanda X menunjukkan register index akan digunakan, simpan akumulator dalam lokasi yg diberikan oleh +5

Catatan :

1. LDN 6 : menghasilkan bilangan 6 yg sedang dimuatkan dengan akumulator
2. ADD @5 : menghasilkan alamat dari data yg sedang dicari dalam lokasi 5

BAHASA TINGKAT RENDAH



- Bahasa yg berorientasi mesin, dimana tiap instruksi sama atau menyerupai instruksi mesin.

Fasilitas umum bahasa tingkat rendah :

1. Kode MNEMONICS digunakan sebagai pengganti kode mesin
2. Alamat simbolis seringkali digunakan sebagai pengganti alamat mesin sebenarnya

Bahasa tingkat rendah harus diterjemahkan ke dalam bahasa mesin sebelum digunakan, sebab walau programmer lebih mudah bekerja dengannya namun ia tidak dapat digunakan oleh mesin dalam bentuk simbolis/mnemonics



Bahasa assembly

- a. Tiap pabrikan komputer biasanya membuat bahasa tingkat rendah yg mendekati kesesuaiannya dengan bahasa mesin tertentu yg digunakan pabrikan tersebut, bahasa ini disebut **bahasa assembly**. Pabrikan tersebut memberikan suatu program yg disebut **assembler** atau **assembler program** yg menterjemahkan **assembly language** ke dalam **machine code**.
- b. Program yg ditulis dalam assembly language disebut **source program**, program yg telah diterjemahkan dalam machine code disebut **object program**.

Bahasa psoudo-assembly

Bahasa assembly yg tergantung mesin yg dibuat utk tujuan pendidikan

MASALAH ILUSTRATIF DALAM BAHASA ASSEMBLY



Format instruksi :

Label : function code operand or operand address

Misal A : LDA 5

Label bersifat opsional dan dapat digunakan utk memberi instruksi alamat simbolis atau memberi nilai numerik suatu alamat simbolis

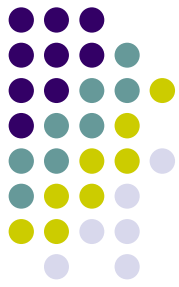
Alamat operand bisa mengambil bentuk :

1. Alamat langsung operand, misal 5
2. Alamat indeks langsung, misal 5,X
3. Alamat tak langsung, misal @5,X

Contoh :

LDA 5 : memuati akumulator dengan isi

LDA 5,X : memuati isi lokasi 5 + (X) dimana (X) berarti isi dari register indeks



SUBROUTINE

- Set instruksi program yg membentuk bagian suatu program dan digunakan utk menjalankan tugas tertentu
- Ada 2 cara umum utk kembali dari subroutine ketika tidak ada fasilitas hardware eksplisit, yaitu dengan cara menggunakan pengalamatan tak langsung atau dengan pemodifikasian alamat.
- Subroutine **open** dan **close**
 - **Subroutine open** adalah bagian dari program utama dan disisipkan atau dimasukkan ke dalam program bila diperlukan
 - **Subroutine close** bukan bagian dari program utama. Ia dihubungkan ke program utamadenan prosedur entri dan return

INSTRUKSI MAKRO



- Instruksi tunggal yg dituliskan sebagai bagian dari bahasa program sumber, yg bila digabungkan akan menggenerasi banyak instruksi kode mesin.

Contoh :

makro yg menambah 1 ke isi lokasi N

MACRO INSTRUCTION ditulis sebagai ADD 1 (N)

ADD 1 (N) generasi S : 0 menyimpan isi akumulator pd saat itu

STA S

LDN 1

ADD N meng ADD N

STA N menambah 1 ke lokasi N

LDA S mengganti isi akumulaotr asli