
Bahasa rakitan/Assembly Language



Tata Sumitra M.Kom

Website: www.tatasumitra.com

Email: tsumitra@gmail.com



Penilaian

- Absen 10 % + Tugas 20 % + Mid 30%
+ UAS 40% = NA 100 %

$80 < NA \leq 100$, grade A

$68 < NA \leq 80$, grade B

$55 < NA \leq 68$, grade C

$34 < NA \leq 55$, grade D

$NA \leq 34$, grade E



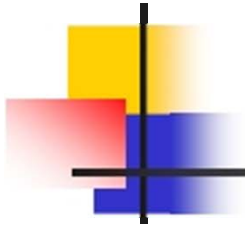
Daftar Pustaka

- Pemrograman dgn Bhs Assembly, Ediman Lukito, 1995, Elex Media, Jakarta
- DOS Assembly Language Programming, Alan R. Miller, Socorro New Mexico, 1988
- Assembly Language 4 The PC, John Socha and Peter Norton, Brady Publishing, New York, Third Edition
- Belajar Sendiri Pemrog. Dgn Bhs. Assembly, Susanto, 1995, Elex Media, Jakarta

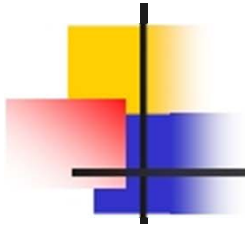


Silabus

- Pengantar Bahasa Rakitan
- DOS
- Register
- Directive, Opcode, Komentar
- Addressing Mode
- Interrupt 20H
- Interrupt 21H service 4 CH



-
- Mnemonic MOV
 - LOOP
 - INC
 - DEC
 - Operasi aritmatika
 - Instruksi Logika

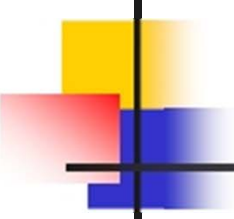


- Instruksi Flow Control
- Inputan Keyboard
- Tampilan Angka
- STACK
- Procedure
- Macro



Pengantar Bahasa Rakitan

- Pendahuluan
- Kelemahan & Kelebihan Bhs Assembly



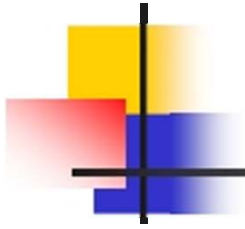
Mengapa perlu belajar bhs assembler??

- Utk lebih memahami tentang arsitektur komputer & mikroprosesor
- Utk menjadi seorang programmer yg baik
- Bhs assembler : bhs pemrog. Terendah yg mendekati bhs mesin
- Membuat sebuah program dpt bekerja lebih cepat & efisien dibanding menggunakan compiler (Pascal atau C/C++)

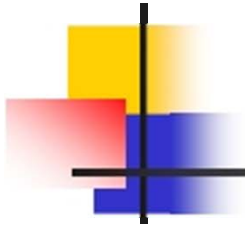


DOS (Disk Operating System)

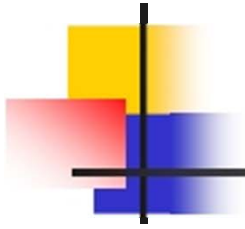
- MD
- CD
- Copy
- DIR
- REN
- DEL
- ERASE
- CLS
- MKDIR
- RMDIR



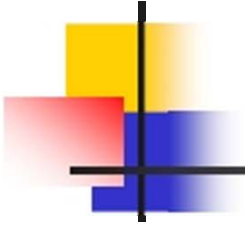
- MASM = Microsoft's Macro Assembler
utk file. Obj
- TASM = Borland's Turbo Assembler



-
- Fact → Komputer hanya dpt menjalankan instruksi yg ditulis dlm sandi biner (berupa 0 atau 1) serta terletak dlm memory utama



- Saat melakukan penulisan program dlm bhs pemrog (BASIC, FOTRAN, dsb) maka scr teoritis telah tertulis prog. sumber dan prog. penterjemah



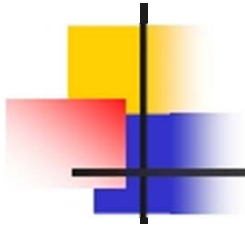
- Bahasa tingkat tinggi menghasilkan prog. Biner yg relatif besar shg tidak cocok utk penciptaan prog. Sistem yg membantu komputer utk berjalan lebih efisien

Kelemahan dan Kelebihan Bhs assembler

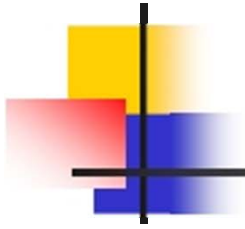


Kelebihan:

- Cocok utk prog. Sistem krn ada korespondensi satu-satu antara source prog. Dgn instruksi biner hasil terjemahannya
- Prog. yg dibentuk merupakan prog kecil shg cara kerja lebih cepat dibanding menggunakan compiler



-
- Assembler → Prog. Penterjemah yg mengubah source prog. Menjadi sandi biner
 - Merupakan bhs tingkat rendah dimana tiap baris prog. Akan sesuai dgn 1 instruksi bahasa mesin



Kelemahan:

- Bahasanya lebih membosankan
- User hrs mengetahui tentang himp. Instruksi CPU
- Penulisan sedikit bertele-tele, beberapa instruksi hanya u/ menjalankan 1 tugas saja



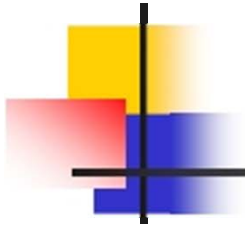
Satuan Bilangan

- Bit (binary digit) = 0 atau 1
- Byte = 8 bit
- Nibble = 4 bit
- Words = 16 bit atau 2 byte



Sistem bilangan

- Biner
- Desimal
- Octal
- Heksadesimal

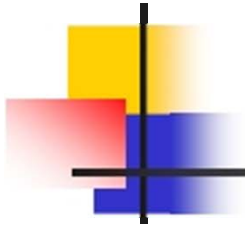


-
- Heksadesimal lebih banyak digunakan saat beroperasi dgn DEBUG



Perintah DEBUG

- Q (quit)
- H (heksadesimal)
- R (register)
- E (enter)
- D (dump)
- A (assemble)
- N (name)
- RCX (register CX)
- RIP (register IP)
- W (write)
- G (go)
- T (trace)
- U (unassemble)



- Q (quit) → utk keluar dr debug
 - H (heksadesimal) → utk menghitung penjumlahan dan pengurangan 2 bil hex
- Format instruksi : H operand1 operand2
- Contoh : H 2204 2012 ↵
4216 (sbg hsl penjum) 01F2 (sbg hsl pengurangan)



- R (register)

Utk menampilkan isi semua register pd saat itu

Contoh R ↵

Ax=0000 BX=0000 CX=0000 DX=0000

SP=CE2E BP=0000 SI=0000 DI=0000

DS=0FD8 ES=0FD8 SS=0FD8 CX=0FD8

IP=0100 NV UP DI PL NZ NA PO NC

0FD8 : 0100 E603 OUT 03,AL

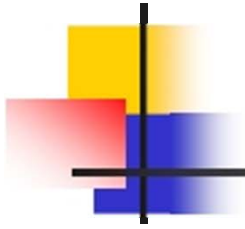


- E (ENTER)

Utk memasukkan data ke alamat yg disebutkan

Format instruksi E alamat [daftar data]

Contoh: E 200 41 42 43 44 ↵



- D (Dump)

Utk menampilkan isi memori tertentu ke layar

Format instruksi : D [range]

- Contoh : D 200 ↵

396F:0200 41 41 43 44 00 30 24.....

396F:0210.....

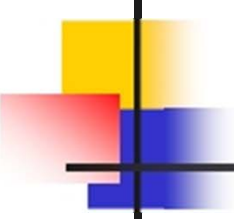


- A (Assemble)

Utk menulis program

Contoh: A 100 ↵

0FD8:0100 ...(isikan baris prog.
Assembly)



- N (name)

Utk memberi nama program

Format instruksi N namafile

- Contoh : N coba.com ↵



- RCX (Register CX)

Utk menampilkan dan mengubah isi register CX yg menyatakan panjang/ukuran program yg sedang ditulis

- Contoh: RCX ↵

CX 0000 (isi panjang program dlm notasi hex)



- RIP (Register IP)

Utk menampilkan dan mengubah isi register IP yg menyatakan alamat awal program yg akan dieksekusi

- Contoh : RIP ↵

IP 0100

:0200 ↵ (jadikan alamat awal=0200H)



- W (Write)

Utk menyimpan (save) program ke disk,
setelah diberikan perintah RIP,RCX dan
N

- Contoh : W ↵

Writing 00008 bytes



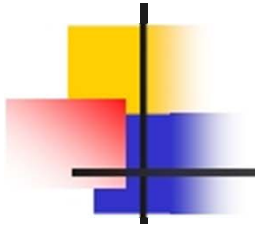
- G (Go)

Utk mengeksekusi prog. Yg sedang aktif

- Contoh : G ↵

Halo

Program terminated normally

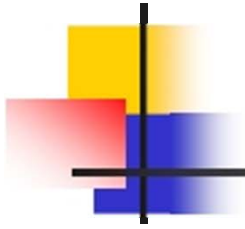


- T (Trace)

Utk mengeksekusi sebaris prog. Dan melihat efeknya terhdp isi register

- Contoh:

T ↙



- U (Unassemble)
- Utk menampilkan list program
- Format instruksi :

U alamat awal \lfloor panjang atau

U [alamat awal] berarti tampilkan program sepanjang 21 H

Contoh : U 100 \lfloor 0D

Membuat prog. Assembly dgn debug

- C:\DOS> debug namafile.com ↵ atau
C:\DOS> debug ↵

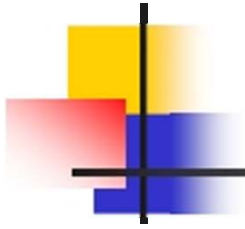
PROG. CETAK DGN INT 21H SERVICE 02H

- A 100

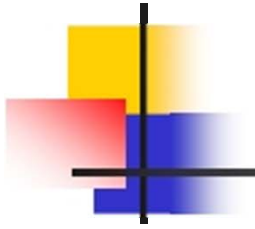
0AD7 : 0100 MOV AH,02 ; MASUKKAN 02 H KE
AH

0AD7 : 0102 MOV DL,48 ; MASUKKAN 48 H KE
DL

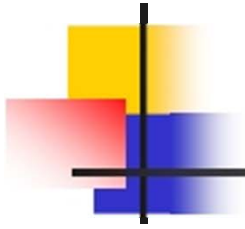
0AD7 : 0104 INT 21H ; INTERRUPT 21H



```
0AD7 : 0106 MOV DL,61 ; CETAK 'a'  
0AD7 : 0108 INT 21H  
0AD7 : 010A MOV DL,6C ; CETAK 'l'  
0AD7 : 010C INT 21H  
0AD7 : 010E MOV DL,6F ; CETAK 'o'  
0AD7 : 0110 INT 21H  
0AD7 : 0112 INT 20H ; HENTIKAN PROG  
0AD7 : 0114 ↵
```



-
- N Coba1.com ↵
 - RCX ↵
 - W ↵
 - RIP ↵
 - G ↵
 - Q ↵



- Lingkungan IBM PC memp. Suatu bagian penting → mikroprosesor / prosesor yg menangani keseluruhan kinerja komputer
- Dasarnya instruksi pada prosesor 8088 masih dipergunakan pd 80286, 80386, 80486, pentium, pentium II, pentium III dan pentium IV

Bahasa Rakitan dalam Sistem Komputer

High Level Language

Application Program

Compiler

Utility

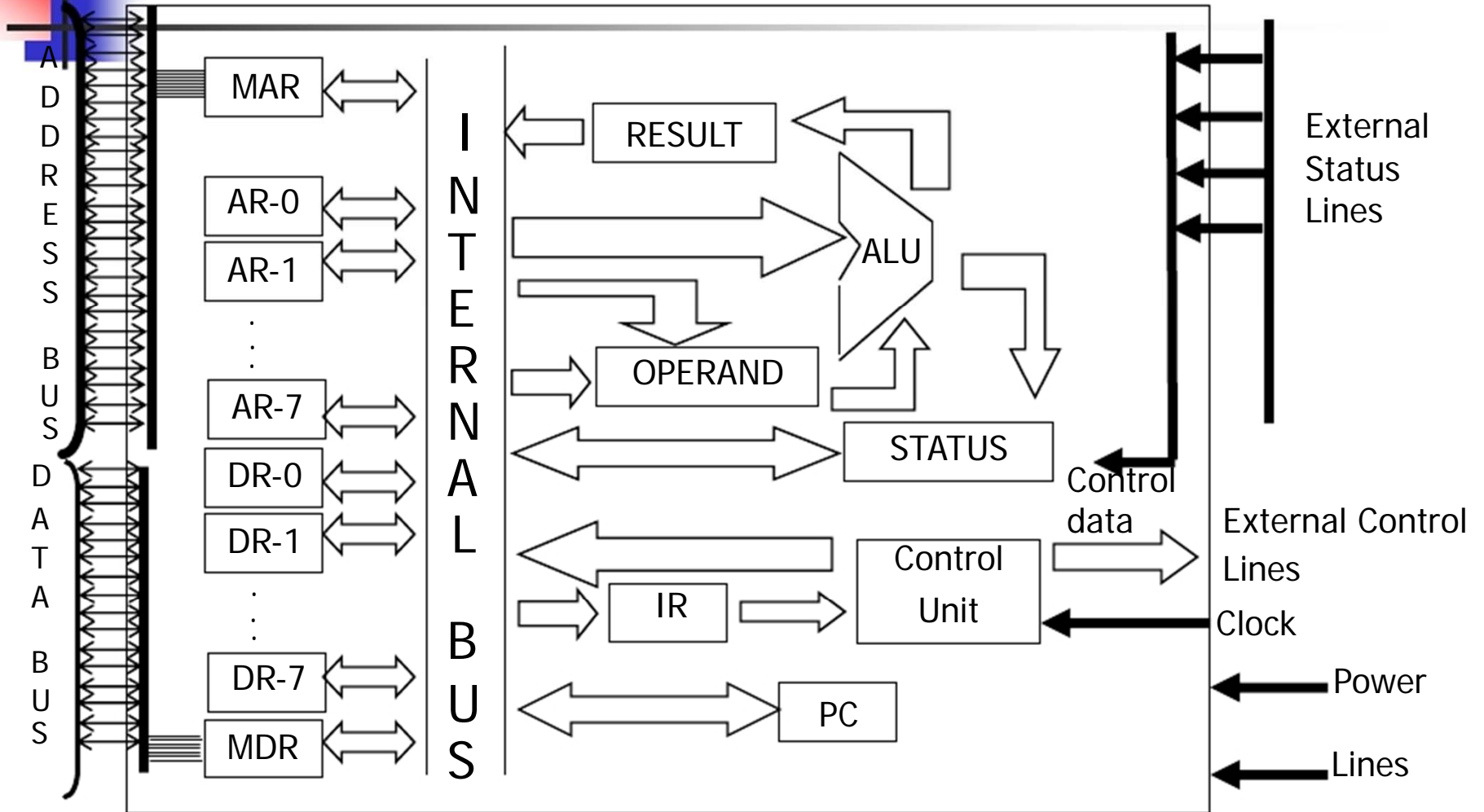
Operating System

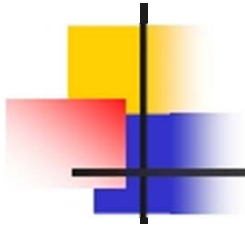
System Programming

Low Level Language

Hardware

Processor





-
- Model (directive) : utk mendefinisikan model memory dr program yg akan dibuat.



Pengalamatan Memori

- Segmen : offset → pengalamatan 16 bit (relative address)
- A123 (segmen) : 099A (offset) = A1BCA (RELATIVE ADDRESS)



Register

- Hampir menyerupai variabel dlm bhs pemrog. Tingkat tinggi. Register merupakan sebagian memori mikroprosesor yg dpt diakses dgn kecepatan yg sangat tinggi



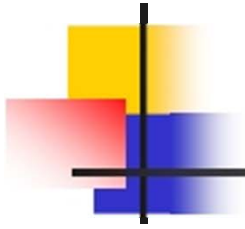
5 Bagian Register

- Segmen Register
- Pointer dan Index Register
- General Purpose Register
- Index Pointer Register
- Flags Register

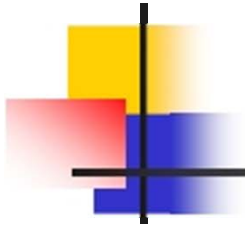


Segmen Register

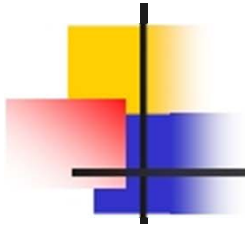
- Terdiri dr CS,DS,ES,SS yg merupakan register 16 bit
- Digunakan utk menunjukkan alamat dr suatu segmen



- CS (Code Segmen) → utk menunjukkan posisi alamat segmen program/kode yg sedang aktif. Berpasangan dgn register IP (CS:IP)
- SS (Stack Segmen) → utk menunjukkan letak dr segmen yg digunakan oleh stack. Berpasangan dgn register SP(SS:SP)



- DS (Data Segmen) → utk menunjukkan posisi segmen data-data program akan disimpan. Umumnya isi register tdk perlu diubah kecuali pd prog. Residen. Berpasangan dgn register DX (DS:DX)

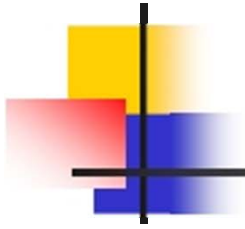


- ES (Extra Segmen) → register bonus yg tdk memp. Tugas khusus. Biasanya digunakan utk menunjukkan suatu alamat di memori, misalkan alamat memori video. Membantu DS pd saat program melakukan operasi ke segmen lain (jika program terdiri dr > 1 segmen)

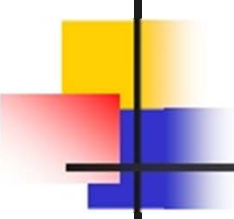


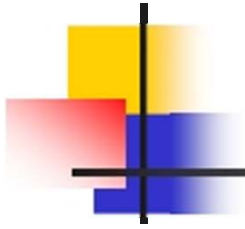
Pointer dan Index Register

- Terdiri dr IP,SP,BP,SI dan DI
- Merupakan register 16 bit
- Digunakan sbg penunjuk atau pointer terhdp suatu lokasi di memori

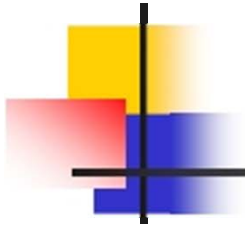


- IP (Instruction Pointer) → utk menunjukkan posisi baris perintah program. Pd saat prog. Baru saja dieksekusi, isinya = alamat awal program
- SP (Stack Pointer) → berpasangan dgn segmen register SS (SS:SP) utk menunjukkan alamat offset dr top of stack, yaitu lokasi item terakhir yg disimpan dlm stack

- 
-
- BP (Base Pointer) → utk mencatat suatu alamat di memori tempat data
 - Register lain utk baca/tulis langsung ke memori selain BX
 - Bedanya BX berpasangan dgn DS (DS:BX) dan BP berpasangan dgn segmen register SS (SS:BP)
 - Digunakan dlm interfacing dgn bahasa lain, misal PASCAL, C



-
- SI (Source Index) dan DI (Destination Index) → digunakan pd operasi string dgn mengakses scr langsung pd alamat di memori yg ditunjukkan oleh kedua register ini

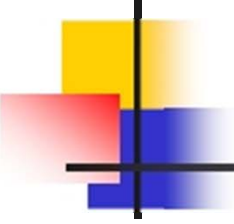


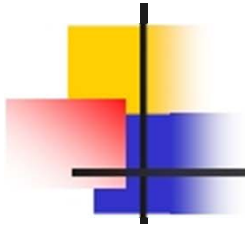
- SI → register u/ menyimpan alamat offset awal memori yg akan dibaca dlm operasi string
- DI → register u/ menyimpan alamat offset awal memori yg akan ditulis dlm operasi string



General Purpose Register

- Terdiri dr AX,BX,CX dan DX
- Terdiri dari 16 bit yg dpt dipisah menjadi 2 bagian (High → 8 bit dan Low → 8 bit)
- Digunakan utk berbagai keperluan

- 
-
- AX (Accumulator) → digunakan pd operasi aritmatika terutama dlm operasi pembagian dan pengurangan. AH digunakan utk menyimpan nomor service INT
 - BX (Base) → digunakan utk baca/tulis memori secara langsung
 - CX (Counter) → digunakan menyimpan hitungan loop



- DX (Data) → utk menyimpan nomor port pd operasi I/O port
 - Offset dr DS dlm pengalamatan memori
 - Membantu AX dlm proses * dan / thdp bilangan 16 bit (menampung sisa hasil pembagian dan hasil word tinggi perkalian)



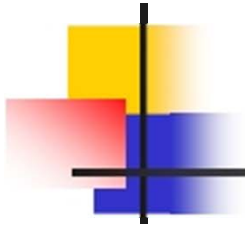
Index Pointer Register

- Register IP berpasangan dgn CS (CS:IP) menunjukkan alamat dr memori tempat dr instruksi selanjutnya yg akan dieksekusi.
- Termasuk register 16 bit



Flag Register

- Menunjukkan kondisi dr suatu keadaan (ya atau tidak)
- Karena setiap keadaan dpt digunakan 1 bit saja, maka sesuai dgn jumlah bitnya
→ flag register mampu mencatat hingga 16 keadaan



- Flag yg terdpt pd mikroprosesor 8088 ke atas terdiri dr 11 FLAG :
OF,SF,ZF,CF,PF,DF,IF,TF,AF,NT,IOPL
- Pd prosesor 80286 dan 80386 ke atas terdpt beberapa tambahan flag yaitu:
PE,MP,EM,TS,ET,RF,VF