

```

<a name="awal">
</a><a href="#akhir">[menuju akhir]</a>
<div style="text-align: center;">
<a href="#">[KEMBALI KE MENU SEBELUMNYA]</a></div>
<br />
<center>
<div style="background-color: white; border: 2px dashed rgb(23, 128, 221); height: 240px; overflow:
auto; padding: 10px; text-align: center; width: 330px;">
<b>DAFTAR ISI</b>
<br />
<div style="text-align: left;">
<a href="#Tujuan">1. Tujuan</a></div>
<div style="text-align: left;">
<a href="#Alat">2. Alat dan Bahan</a></div>
<div style="text-align: left;">
<a href="#Dasar">3. Dasar Teori</a></div>
<div style="text-align: left;">
<a href="#Percobaan">4. Percobaan</a><br />
<div style="text-align: left;">
<a href="#Video">5. Video Simulasi</a></div>
<div style="text-align: left;">
<a href="#Download">6. Download File</a>&nbsp;</div>
<div style="text-align: left;">
<a href="#link"></a></div>
</div>
</div>
</center>

```

```

<p>&nbsp;<a name="Tujuan"></a>1. Tujuan&nbsp;<a href="#">[Kembali]</a></p><p></p><p></p><ul
style="text-align: left;"><li><span style="font-family: times;">Mampu memahami dan membuat
bahasa assambler menggunakan emu8086</span></li><li><span style="font-family: times;">Dapat
membuat program untuk menulis dan membaca input bahasa assambler</span></li></ul><p><a
name="Alat"></a>2. Alat dan Bahan&nbsp;<a href="#">[Kembali]</a></p><p></p><p></p><ul

```

[illegible]

Purpose Register dapat dibagi menjadi register low dan high bits, yang masing-masing

berkemampuan 8 bit. Seperti register AX, register low-nya adalah AL dan register high-nya adalah AH.

Ada empat macam General Purpose Register yaitu:

- a. Register AX (AX+AL) atau Accumulator Register adalah register aritmatik untuk dipakai dalam operasi penambahan, pengurangan, perkalian, dan pembagian. Register AH dapat juga digunakan untuk tempat kembali nilai service number untuk beberapa interrupt tertentu.
- b. Register BX (BH+BL) atau Base Register adalah salah satu dari dua register base Addressing mode, yang dapat mengambil atau menulis langsung dari/ke memori.
- c. Register CX (CH+CL) atau Counter Register adalah suatu counter untuk meletakkan jumlah lompatan pada loop yang dilakukan.
- d. Register DX (DH+DL) atau Data Register mempunyai tiga tugas, yaitu:

1. Membantu AX dalam proses perkalian dan pembagian, terutama perkalian dan pembagian 16 bit.
2. DX merupakan register offset dari DS.
3. DX bertugas menunjukkan nomor port pada operasi port.

2. Segment Register

- a. CS (Code Segment) Register berfungsi untuk menunjukkan segment, program yang berapa dengan pasangan register ini adalah register IP.
- b. DS (Data Segment) Register berfungsi untuk menunjukkan segment dari segment data. Pasangan dari DS adalah DX.
- c. SS (Stack Segment) Register berfungsi untuk menunjukkan segment dari segment stack. Pasangan dari

SS adalah SP.

d. ES (Extra Segment) Register berfungsi untuk pemograman pada saat melakukan operasi ke segment lain.

3. Pointer Register berfungsi untuk menyimpan offset dari relative address

a. IP (Instruction Pointer) Register berfungsi untuk menunjukkan baris perintah program. Pada saat program pertama dijalankan register akan langsung menunjuk pada awal program.

b. SP (Stack Pointer) Register adalah merupakan pasangan SS yang digunakan untuk operasi stack. Pada saat pertama program dijalankan register ini akan menunjuk pada byte terakhir stack.

c. BP (Base Pointer) Register mempunyai fungsi yang sama dengan register BX tetapi BX untuk menulis dan membaca segment DS (Data Segment) sedangkan BP menulis dan membaca segment SS (Stack Segment).

4. Index Register yaitu SI (Source Index) dan DI (Destination Index) berfungsi untuk melakukan operasi

STRING. Namun demikian, kedua register ini sering digunakan untuk menulis dan membaca dari atau ke memori seperti halnya BX dan BP.

5. Flag Register berfungsi untuk mengecek apakah program berfungsi atau tidak. Contohnya :

a. Interrupt Flag, untuk mengecek apakah pada saat operasi Interrupt sedang aktif atau tidak, jika tidak aktif maka Interrupt tidak akan dijalankan.

b. Carry Flag, untuk mengecek apakah saat operasi terjadi kesalahan atau tidak.

c. Sign Flag, untuk menunjukkan apakah suatu bilangan bertanda atau tidak dan sebaliknya.

B. PERINTAH DASAR ASSEMBLER

1. MOV

Adalah perintah untuk mengisi, memindahkan atau memperbaharui isi suatu register, variabel atau suatu lokasi memori.

Syntax : MOV[operand1],[operand2]

a. operand1 berupa register, variabel, lokasi memori

b. operand2

berupa register, variabel, lokasi memori ataupun bilangan

Contoh

MOV AH,AL ; mengkopi isi register AL ke register AH

MOV AH,02 ; mengisi register AH dengan 02

2. Int (Interrupt)

Subrutine yang akan dipanggil setelah tersedia pada memori komputer.

Ada dua jenis yaitu :

a. BIOS Interrupt yaitu 0H hingga 1FH yang disediakan oleh BIOS (Basic Input Output System).

Contoh : Int 16H service 1FH berfungsi untuk mengecek apakah ada tombol keyboard yang ditekan.

b. DOS Interrupt yaitu Int 1FH keatas yang disediakan oleh DOS (Disk Operating System).

Contoh : Int 20H berfungsi menghentikan kerja suatu program.

C. Int 21h service 0Ah

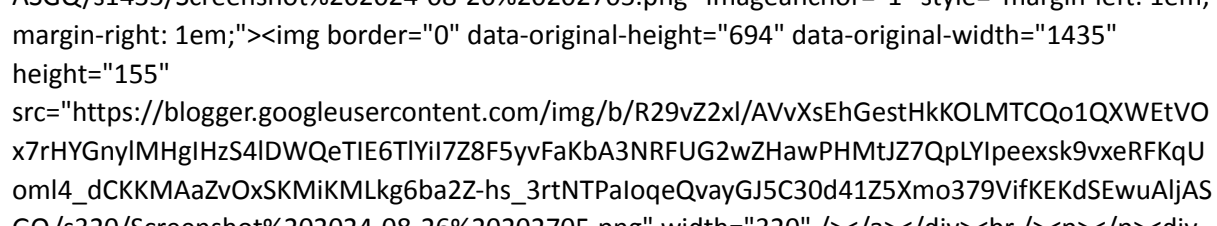
Int 21h adalah salah satu Int yang termasuk DOS Interrupt, karena INT 21h mempunyai banyak sekali tugas, maka tugasnya dibagi-bagi menjadi beberapa bagian. Untuk memanggil bagian-bagian itu perlu menyertakan nomor bagiannya yang disebut juga service number. Contoh 21h yang bertugas mencatat sebuah huruf ke layar yaitu fungsi kedua dari Int 21h.

Service 0AH digunakan untuk melakukan input dengan lebih dari 1 karakter, dengan syarat register harus berisikan service number dari 21H tersebut.

Service 09H digunakan untuk mencetak huruf ke layar, untuk penggunaannya diakhiri dengan tanda dolar '\$'

4. Percobaan

[Kembali]



https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEhGestHkKOLMTCQo1QXWetVOx7rHYGnyIMHglHzS4lDWQeTIE6TIYil7Z8F5yvFaKbA3NRFUG2wZHawPHMtJZ7QpLYlIpeexsk9vxeRFKqUoml4_dCKKMAaZvOxSKMiKMLkg6ba2Z-hs_3rtNTPaloqeQvayGJ5C30d41Z5Xmo379VifKEKdSEwuAljASGQ/s320/Screenshot%202024-08-26%20202705.png

xjhRri4O7K1KKuawlb3g8kUH7a9jpFVN8VB5Rt9h0WGSwLvKJXfKDh9buLQYIGmxLbafiYt_-5Ac0ielF8Cw6oo/s849/ss%20mikro%20tgs%201.png" imageanchor="1" style="margin-left: 1em; margin-right: 1em;"></div><p></p><div class="separator" style="clear: both; text-align: center;"></div>
<div class="separator" style="clear: both; text-align: center;"></div>
<div class="separator" style="clear: both; text-align: center;"></div><p>
</p><div>Listing Program :</div><div>
</div><div><p>PROGRAM MENULIS DAN MEMBACA MEMORI</p><p>;Program isikan Nomor Induk Mahasiswa (NIM)</p><p> Start: JMP MULAI</p><p> TANYA db 'NIM Anda?:',13,10, '\$'</p><p> NIM db 11,?,11 DUP(?)</p><p> JAWAB db 13,10, 'NIM:\$'</p><p> MULAI:

MOV AH,09H

DX, OFFSET TANYA

INT 21H

TANYA

MOV AH,0AH

MOV DX, OFFSET NIM

PUSH DX

MENYIMPAN DX KE STACK

INT 21H

INPUT DARI KEYBOARD

MOV BX,OFFSET NIM

INC BX

MENUNJUK KEPADA INDIKATOR (mm)

BEBERAPA HURUF YANG TELAH DIMASUKKAN

MOV DL,[BX]

MASUKKAN [BX] KE DX

XOR DH,DH

INC BX

menunjuk kepada awal kata yang dimasukkan

ADD BX,DX

supaya menunjuk kepada huruf terakhir masukan

MOV DL,'\$'

mengganti ODh dengan '\$' supaya bisa dicetak

MOV [BX],DL

MASUKAN DL KE [BX]

MOV DX, OFFSET JAWAB

MOV AH,09H

INT 21H

CETAK JAWAB

POP DX

AMBIL DX DARI STACK

INC DX

menambahkannya dengan 2 supaya menunjuk kepada awal masukan

MOV AH,09H

CETAK TAMPUNG

INT 21H

MOV AH,4CH

INT 21H

5. Video Simulasi

[\[Kembali\]](#)

[Download](#)

6. Download File

[\[Kembali\]](#)

Download program

https://drive.google.com/file/d/1W_kq7L2gDdlG2zg26ZktPbrvKGAHyMgs/view?usp=drive_link

" style="color: #cc6611; text-decoration-line: none;" target="_blank">[klik disini]
</p><p style="background-color: white; color: #222222;"> b. Video penjelasan [klik disini]</p><p style="background-color: white; color: #222222;"> c. Datasheet Intel 8086 [klik disini]</p><p style="background-color: white; color: #222222;"> e. Program Emu8086 [klik disini]</p><p style="background-color: white; color: #222222;"> f. Kode HTML [klik disini]</p><p>
</p>

<div style="text-align: center;"></div>[menuju awal]<a>

