

LAPORAN AKHIR

Mata Praktikum : Organisasi Sistem Komputer
Kelas : 2IA16
Praktikum ke - : 3
Tanggal : Kamis, 28 Oktober 2021
Materi : Addressing
NPM : 50420093
Nama : Ajay Alfredo Almani
Ketua Asisten : Brigitta
Nama Asisten :
Paraf Asisten :
Jumlah Lembar : 5 Lembar



LABORATORIUM TEKNIK INFORMATIKA
UNIVERSITAS GUNADARMA
2021

LISTING PROGRAM

```
1  .MODEL SMALL
2  .CODE
3  ORG 100h
4
5  mulai: jmp proses
6  nilA db 55
7  nilB db 55
8
9  kat0 db 'nilai a lebih kecil dari nilai b.$'
10 kat1 db 'nilai a sama dengan nilai b.$'
11 kat2 db 'nilai a lebih besar dari nilai b.$'
12
13 proses:
14     mov al, nilA
15     cmp al, nilB
16     JB Akecil; bil A < bil B
17     JE sama; bil A = bil B
18     JA Abesar; bil A > bil B
19
20 Akecil:
21     lea dx, kat0
22     jmp cetak
23
24 sama:
25     lea dx, kat1
26     jmp cetak
27
28 Abesar:
29     lea dx, kat2
30     jmp cetak
31
32 cetak:
33     mov ah, 09h
34     int 21h
35     int 20h
36
37 end mulai
```

Assembly language source file length : 498 lines : 37 Ln : 37 Col : 10 Pos : 499 Windows (CR LF) UTF-8 INS

LOGIKA PROGRAM

Pada Pertemuan Praktikum pertama ini yaitu Organisasi Sistem Komputer, Saya mendapat materi Addressing Turbo Assembler (TASM) merupakan perangkat lunak bahasa pemrograman Assembly yang dikembangkan Borland untuk komputer personal yang kompatibel dengan IBM, umumnya berbasis prosesor x86. Perangkat lunak ini diperdagangkan oleh Borland sebagai sebuah produk mandiri, atau dipaketkan bersama produk perangkat lunak bahasa pemrograman tingkat tinggi yang juga dikembangkan Borland, biasanya produk untuk tingkat mahir seperti (Borland Pascal, atau Borland C++), dan memiliki integrasi yang sangat baik dengan bahasa-bahasa pemrograman tingkat tinggi tersebut. Dengan berakhirnya era kejayaan keluarga Turbo, Turbo Assembler saat ini sudah tidak lagi dikembangkan oleh Borland. Dibawah ini kita akan menampilkan nilai a sama dengan nilai b.

```
1 .MODEL SMALL
2 .CODE
3 ORG 100h
4
5 mulai: jmp proses
6 nilA db 55
7 nilB db 55
8
```

Langkah pertama pada gambar atas yaitu kita menuliskan **.MODEL SMALL** yang merupakan keterangan program yang berfungsi untuk mode dari program yang sudah ada. **.CODE ORG 100h** yang merupakan keterangan yang digunakan untuk menentukan alamat offset terhadap program yang akan digunakan. Perintah **JMP(JUMP)** yang berfungsi untuk melompat menuju tempat yang di tunjukkan oleh perintah JUMP. Code nya adalah **JUMP proses**. Selanjutnya kita membuat variabel **nilA** dan **nilB** sekaligus memasukkan nilainya. **nilA** dan **nilB** yang merupakan variabel nilai 1 dan nilai 2, lalu kita masukkan nilainya **55**. Disini saya membuat kedua nilai sama dengan tujuan untuk menampilkan output “**nilai a sama dengan nilai b**”.

```
8
9 kat0 db 'nilai a lebih kecil dari nilai b.$'
10 kat1 db 'nilai a sama dengan nilai b.$'
11 kat2 db 'nilai a lebih besar dari nilai b.$'
12
13 proses:
14     mov al, nilA
15     cmp al, nilB
16     JB Akecil ; bil A < bil B
17     JE sama ; bil A = bil B
18     JA Abesar ; bil A > bil B
19
```

Langkah Selanjutnya pada gambar atas yaitu berfungsi untuk membuat keterangan yang akan ditampilkan pada output program. Disini kita membuat 3 variabel, yaitu **kat0**, **kat1**, dan **kat2**. **Kat0** akan tampil jika **nilai a lebih kecil dari nilai b**, **Kat1** akan tampil jika **nilai a sama dengan nilai b**, dan **Kat2** akan tampil jika **nilai a lebih besar dari nilai b**. Selanjutnya pada bagian **proses** berfungsi untuk membuat proses agar program berjalan sesuai syarat yang telah ditentukan. **Mov** adalah perintah untuk mengisi nilai ke register, variabel, atau alamat memory tertentu. **mov al, nilA** adalah **nilai al akan berisi nilai dari variabel nilA**. **CMP** digunakan untuk membuat perbandingan antara pernyataan satu dan pernyataan lainnya. Disini cmp digunakan untuk membandingkan **nilA dengan nilB**. Kemudian terdapat kode **JB**, **JE**, dan **JA**. **JB (Jump If Below)**, lompat jika **Operand1 < Operand2**, **JE (Jump If Equal)**, lompat jika **Operand1 = Operand2**, **JA (Jump If Above)**, lompat jika **Operand1 > Operand2**. **Operand1** merupakan **bil A** dan **Operand2** merupakan **bil B**.

```

19
20 Akecil:
21     → lea dx, kat0
22     → jmp cetak
23
24 sama:
25     → lea dx, kat1
26     → jmp cetak

```

Langkah Selanjutnya pada gambar atas yaitu berfungsi untuk mencetak keterangan sesuai dengan hasil perbandingan kedua nilai. Bagian ini terbagi menjadi 3 label, yaitu **Akecil:** untuk **lebih kecil dari**, **sama:** untuk **sama dengan**.

```

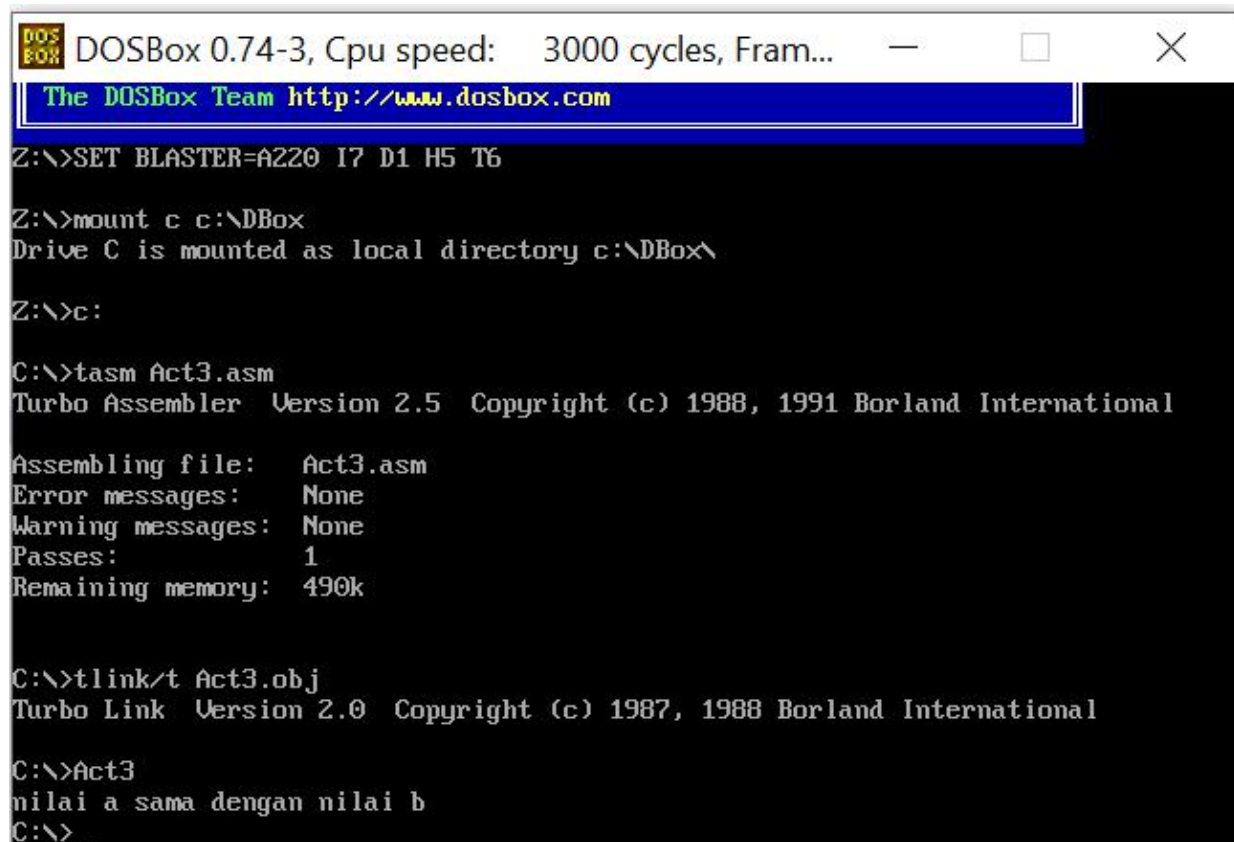
27
28 Abesar:
29     → lea dx, kat2
30     → jmp cetak
31
32 cetak:
33     → mov ah, 09h
34     → int 21h
35     → int 20h
36
37 end mulai

```

Assembly language source file length : 498 lines : 37 Ln : 37 Col : 10 Pos : 499 Windows (CR LF) UTF-8 INS

Langkah Selanjutnya pada gambar terakhir di atas yaitu **Abesar:** untuk **lebih besar dari**. Pada program di atas yaitu hanya menggunakan **lea** dan **jmp**. LEA adalah perintah untuk mendapatkan alamat dari sebuah variabel. Programnya adalah LEA variabel. Berdasarkan contoh diatas yaitu **lea dx, kat0** akan mengambil variabel **kat0** yang telah dibuat sebelumnya. Kemudian **jmp cetak** untuk melakukan lompatan ke **cetak**. Begitu juga dengan bagian **sama** dan **Abesar**;, yang membedakan hanya variabel yang dipanggil untuk mencetak keterangan. Register **dx** harus berisi alamat dari string yang akan dicetak. Selanjutnya membuat **cetak** yang menggunakan sintaks **mov ah, 09h** yaitu akan mencetak karakternya, **int 21h** dan **int 20h** yaitu sebagai penutup program.

OUTPUT PROGRAM



The screenshot shows a DOSBox window titled "DOSBox 0.74-3, Cpu speed: 3000 cycles, Fram...". The window has a blue title bar with the DOSBox logo and a URL "The DOSBox Team http://www.dosbox.com". The main area is black with white text. The commands and output are as follows:

```
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c c:\DBox
Drive C is mounted as local directory c:\DBox\

Z:\>c:

C:\>tasm Act3.asm
Turbo Assembler Version 2.5 Copyright (c) 1988, 1991 Borland International

Assembling file:   Act3.asm
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  490k

C:\>tlink/t Act3.obj
Turbo Link Version 2.0 Copyright (c) 1987, 1988 Borland International

C:\>Act3
nilai a sama dengan nilai b
C:\>
```