

# **Laporan Praktikum Sistem Operasi**

## **Modul 1**



Dosen Pengampu : Heru Setya Nugraha, S.T., M.Kom.

Disusun Oleh :

Nama : Dika Pramudyatama

NIM : L200190224

Program Studi : Informatika

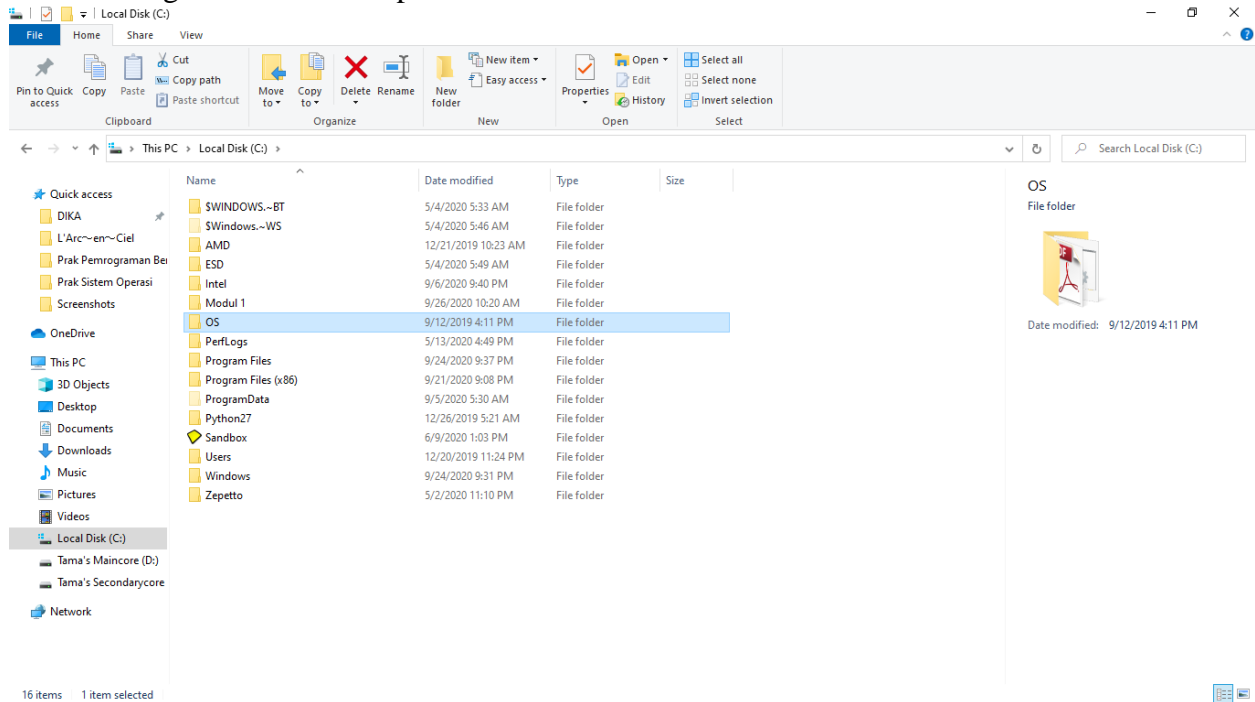
**Universitas Muhammadiyah Surakarta**

**2020**

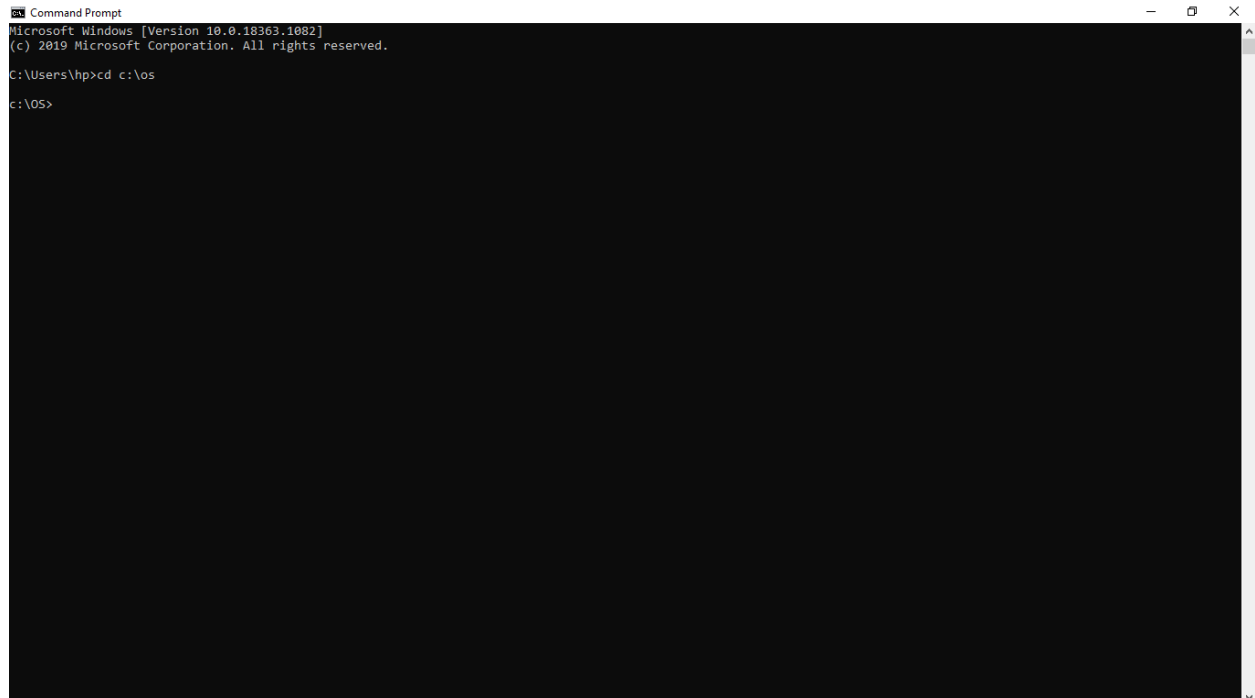
# Modul 1

## Pengenalan Sistem Pengembangan OS dengan PC Simulator 'Bochs'

### 1. Mengextract file OS.zip ke disk C



### 2. Membuka folder OS di Command Prompt



### 3. Menuliskan perintah 'dir' untuk melihat isi folder OS

```
Command Prompt
Microsoft Windows [Version 10.0.18363.1882]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\hp>cd c:\os

c:\os>dir
Volume in drive C has no label.
Volume Serial Number is 76D5-D97F

Directory of c:\os

09/12/2019  04:11 PM    <DIR>          .
09/12/2019  04:11 PM    <DIR>          ..
09/12/2019  04:11 PM    <DIR>          Bochs-2.3.5
09/12/2019  04:11 PM    <DIR>          Dev-Cpp
12/17/2008  12:08 AM             1,096,291  1386.pdf
09/12/2019  04:11 PM    <DIR>          LAB
12/17/2008  12:07 AM             846,920  pcasm-book.pdf
12/17/2008  01:44 AM              86  Setpath.bat
12/13/2008  02:12 PM             716,512  winima81.exe
               4 File(s)          2,659,809 bytes
               5 Dir(s)         43,597,590,528 bytes free

c:\os>
```

### 4. Menuliskan perintah 'setpath' untuk menjalankan setpath

```
Command Prompt
Microsoft Windows [Version 10.0.18363.1882]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\hp>cd c:\os

c:\os>dir
Volume in drive C has no label.
Volume Serial Number is 76D5-D97F

Directory of c:\os

09/12/2019  04:11 PM    <DIR>          .
09/12/2019  04:11 PM    <DIR>          ..
09/12/2019  04:11 PM    <DIR>          Bochs-2.3.5
09/12/2019  04:11 PM    <DIR>          Dev-Cpp
12/17/2008  12:08 AM             1,096,291  1386.pdf
09/12/2019  04:11 PM    <DIR>          LAB
12/17/2008  12:07 AM             846,920  pcasm-book.pdf
12/17/2008  01:44 AM              86  Setpath.bat
12/13/2008  02:12 PM             716,512  winima81.exe
               4 File(s)          2,659,809 bytes
               5 Dir(s)         43,597,590,528 bytes free

c:\os>setpath

c:\os>Path=C:\os\Dev-Cpp\bin;C:\os\Bochs-2.3.5;c:\os\Perl;c:\Windows;c:\Windows\System32
c:\os>
```

## 5. Menuju ke folder LAB1 dengan perintah 'cd lab\lab1'

```
Command Prompt
Microsoft Windows [Version 10.0.18363.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\hnp>cd c:\os

c:\os>dir
Volume in drive C has no label.
Volume Serial Number is 76D5-D97F

Directory of c:\os

09/12/2019  04:11 PM  <DIR>          .
09/12/2019  04:11 PM  <DIR>          ..
09/12/2019  04:11 PM  <DIR>          Bochs-2.3.5
09/12/2019  04:11 PM  <DIR>          Dev-Cpp
12/17/2008  12:08 AM             1,096,291  1386.pdf
09/12/2019  04:11 PM  <DIR>          LAB
12/17/2008  12:07 AM             846,920  pcasm-book.pdf
12/17/2008  01:44 AM              86  Setpath.bat
12/13/2008  02:12 PM             716,512  winima81.exe
               4 File(s)      2,659,809 bytes
               5 Dir(s)      43,597,590,528 bytes free

c:\os>setpath

c:\os>Path=C:\OS\Dev-Cpp\bin;C:\OS\Bochs-2.3.5;c:\OS\Perl;c:\Windows;c:\Windows\System32
c:\os>cd lab\lab1

c:\OS\LAB\LAB1>
```

## 6. Menuliskan perintah 'dir' untuk melihat isi folder LAB1

```
Command Prompt

c:\os>dir
Volume in drive C has no label.
Volume Serial Number is 76D5-D97F

Directory of c:\os

09/12/2019  04:11 PM  <DIR>          .
09/12/2019  04:11 PM  <DIR>          ..
09/12/2019  04:11 PM  <DIR>          Bochs-2.3.5
09/12/2019  04:11 PM  <DIR>          Dev-Cpp
12/17/2008  12:08 AM             1,096,291  1386.pdf
09/12/2019  04:11 PM  <DIR>          LAB
12/17/2008  12:07 AM             846,920  pcasm-book.pdf
12/17/2008  01:44 AM              86  Setpath.bat
12/13/2008  02:12 PM             716,512  winima81.exe
               4 File(s)      2,659,809 bytes
               5 Dir(s)      43,597,590,528 bytes free

c:\os>setpath

c:\os>Path=C:\OS\Dev-Cpp\bin;C:\OS\Bochs-2.3.5;c:\OS\Perl;c:\Windows;c:\Windows\System32
c:\os>cd lab\lab1

c:\OS\LAB\LAB1>dir
Volume in drive C has no label.
Volume Serial Number is 76D5-D97F

Directory of c:\OS\LAB\LAB1

09/24/2020  11:16 PM  <DIR>          .
09/24/2020  11:16 PM  <DIR>          ..
09/24/2020  11:18 PM             10,840  bochsout.txt
12/15/2008  04:17 PM             1,628  bochsrc.bxrc
12/14/2008  12:02 PM             14,365  boot.asm
09/24/2020  11:11 PM              512  boot.bin
09/16/2015  07:51 AM              512  boots.bin
12/15/2008  12:47 AM              78  dosfp.bat
09/24/2020  11:17 PM          1,474,560  floppy.img
12/14/2008  11:45 AM              7,966  kernel.asm
12/15/2008  04:21 PM              227  Makefile
12/15/2008  12:20 PM              44  s.bat
               10 File(s)      1,510,732 bytes
               2 Dir(s)      43,603,685,376 bytes free

c:\OS\LAB\LAB1>
```

## 7. Menuliskan perintah 'del floppy.img' untuk menghapus file floppy.img

```
Command Prompt
Volume in drive C has no label.
Volume Serial Number is 76D5-D97F

Directory of c:\OS

09/12/2019  04:11 PM  <DIR>          .
09/12/2019  04:11 PM  <DIR>          ..
09/12/2019  04:11 PM  <DIR>          Bochs-2.3.5
09/12/2019  04:11 PM  <DIR>          Dev-Cpp
12/17/2008  12:08 AM             1,096,291  1386.pdf
09/12/2019  04:11 PM  <DIR>          LAB
12/17/2008  12:07 AM             846,920  pcasm-book.pdf
12/17/2008  01:44 AM              86  Setpath.bat
12/13/2008  02:12 PM             716,512  winima81.exe
               4 File(s)          2,659,809 bytes
               5 Dir(s)          43,597,590,528 bytes free

c:\OS>setpath
c:\OS>Path=C:\OS\Dev-Cpp\bin;C:\OS\Bochs-2.3.5;c:\OS\Perl;c:\Windows;c:\Windows\System32
c:\OS>cd lab\lab1

c:\OS\LAB\LAB1>dir
Volume in drive C has no label.
Volume Serial Number is 76D5-D97F

Directory of c:\OS\LAB\LAB1

09/24/2020  11:16 PM  <DIR>          .
09/24/2020  11:16 PM  <DIR>          ..
09/24/2020  11:18 PM             10,840  bochsout.txt
12/15/2008  04:17 PM             1,628  bochsrc.bxrc
12/14/2008  12:02 PM             14,365  boot.asm
09/24/2020  11:11 PM              512  boot.bin
09/16/2015  07:51 AM              512  boots.bin
12/15/2008  12:47 AM              78  dosfp.bat
09/24/2020  11:17 PM          1,474,560  floppy.img
12/14/2008  11:45 AM             7,966  kernel.asm
12/15/2008  04:21 PM              227  Makefile
12/15/2008  12:20 PM              44  s.bat
               10 File(s)          1,510,732 bytes
               2 Dir(s)          43,603,685,376 bytes free

c:\OS\LAB\LAB1>del floppy.img
c:\OS\LAB\LAB1>
```

## 8. Melakukan kompilasi source code program boot.asm dengan perintah 'make fp.disk' lalu akan muncul output boot.bin yang isinya disalin ke floppy.img

```
Command Prompt
12/17/2008  12:08 AM             1,096,291  1386.pdf
09/12/2019  04:11 PM  <DIR>          LAB
12/17/2008  12:07 AM             846,920  pcasm-book.pdf
12/17/2008  01:44 AM              86  Setpath.bat
12/13/2008  02:12 PM             716,512  winima81.exe
               4 File(s)          2,659,809 bytes
               5 Dir(s)          43,597,590,528 bytes free

c:\OS>setpath
c:\OS>Path=C:\OS\Dev-Cpp\bin;C:\OS\Bochs-2.3.5;c:\OS\Perl;c:\Windows;c:\Windows\System32
c:\OS>cd lab\lab1

c:\OS\LAB\LAB1>dir
Volume in drive C has no label.
Volume Serial Number is 76D5-D97F

Directory of c:\OS\LAB\LAB1

09/24/2020  11:16 PM  <DIR>          .
09/24/2020  11:16 PM  <DIR>          ..
09/24/2020  11:18 PM             10,840  bochsout.txt
12/15/2008  04:17 PM             1,628  bochsrc.bxrc
12/14/2008  12:02 PM             14,365  boot.asm
09/24/2020  11:11 PM              512  boot.bin
09/16/2015  07:51 AM              512  boots.bin
12/15/2008  12:47 AM              78  dosfp.bat
09/24/2020  11:17 PM          1,474,560  floppy.img
12/14/2008  11:45 AM             7,966  kernel.asm
12/15/2008  04:21 PM              227  Makefile
12/15/2008  12:20 PM              44  s.bat
               10 File(s)          1,510,732 bytes
               2 Dir(s)          43,603,685,376 bytes free

c:\OS\LAB\LAB1>del floppy.img
c:\OS\LAB\LAB1>make fp.disk
nasm boot.asm -o boot.bin -f bin
dd if=boot.bin of=floppy.img
rawwrite dd for windows version 0.5.
Written by John Newbigin <jn@it.swin.edu.au>
This program is covered by the GPL.  See copying.txt for details
1+0 records in
1+0 records out

c:\OS\LAB\LAB1>
```

## 9. Masukkan perintah 'bximage' untuk membuat file floppy.img baru

```
Command Prompt - bximage
c:\OS>setpath
c:\OS>Path=C:\OS\Dev-Cpp\bin;C:\OS\Bochs-2.3.5;c:\OS\Perl;C:\Windows;C:\Windows\System32
c:\OS>cd lab\lab1

c:\OS\LAB\LAB1>dir
Volume in drive C has no label.
Volume Serial Number is 76D5-D97F

Directory of c:\OS\LAB\LAB1

09/24/2020  11:16 PM  <DIR>      .
09/24/2020  11:16 PM  <DIR>      ..
09/24/2020  11:18 PM             10,840 bochsout.txt
12/15/2008  04:17 PM             1,628 bochsrc.bxrc
12/14/2008  12:02 PM            14,365 boot.asm
09/24/2020  11:11 PM              512 boot.bin
09/16/2015  07:51 AM              512 boots.bin
12/15/2008  12:47 AM              78 dosfp.bat
09/24/2020  11:17 PM          1,474,560 floppypa.img
12/14/2008  11:45 AM             7,966 kernel.asm
12/15/2008  04:21 PM              227 Makefile
12/15/2008  12:20 PM              44 s.bat
               10 File(s)      1,510,732 bytes
               2 Dir(s)      43,603,685,376 bytes free

c:\OS\LAB\LAB1>del floppypa.img

c:\OS\LAB\LAB1>make fp.disk
nasm boot.asm -o boot.bin -f bin
dd if=boot.bin of=floppya.img
rawwrite dd for windows version 0.5.
Written by John Newbigin <jn@it.swin.edu.au>
This program is covered by the GPL. See copying.txt for details
1+0 records in
1+0 records out

c:\OS\LAB\LAB1>bximage
=====
                        bximage
          Disk Image Creation Tool for Bochs
      $Id: bximage.c,v 1.32 2006/06/16 07:29:33 vruppert Exp $
=====

Do you want to create a floppy disk image or a hard disk image?
Please type hd or fd. [hd]
```

## 10. Ketik 'fd' untuk meilih type file fd karena yang lebih kecil

```
Command Prompt - bximage
c:\OS\LAB\LAB1>dir
Volume in drive C has no label.
Volume Serial Number is 76D5-D97F

Directory of c:\OS\LAB\LAB1

09/24/2020  11:16 PM  <DIR>      .
09/24/2020  11:16 PM  <DIR>      ..
09/24/2020  11:18 PM             10,840 bochsout.txt
12/15/2008  04:17 PM             1,628 bochsrc.bxrc
12/14/2008  12:02 PM            14,365 boot.asm
09/24/2020  11:11 PM              512 boot.bin
09/16/2015  07:51 AM              512 boots.bin
12/15/2008  12:47 AM              78 dosfp.bat
09/24/2020  11:17 PM          1,474,560 floppypa.img
12/14/2008  11:45 AM             7,966 kernel.asm
12/15/2008  04:21 PM              227 Makefile
12/15/2008  12:20 PM              44 s.bat
               10 File(s)      1,510,732 bytes
               2 Dir(s)      43,603,685,376 bytes free

c:\OS\LAB\LAB1>del floppypa.img

c:\OS\LAB\LAB1>make fp.disk
nasm boot.asm -o boot.bin -f bin
dd if=boot.bin of=floppya.img
rawwrite dd for windows version 0.5.
Written by John Newbigin <jn@it.swin.edu.au>
This program is covered by the GPL. See copying.txt for details
1+0 records in
1+0 records out

c:\OS\LAB\LAB1>bximage
=====
                        bximage
          Disk Image Creation Tool for Bochs
      $Id: bximage.c,v 1.32 2006/06/16 07:29:33 vruppert Exp $
=====

Do you want to create a floppy disk image or a hard disk image?
Please type hd or fd. [hd] fd

Choose the size of floppy disk image to create, in megabytes.
Please type 0.16, 0.18, 0.32, 0.36, 0.72, 1.2, 1.44, 1.68, 1.72, or 2.88.
[1.44] _
```

## 11. Pilih ukuran 1.44 MB dengan mengetik '1.44'

```
Command Prompt - bximage
09/24/2020 11:18 PM      10,840 bochsout.txt
12/15/2008 04:17 PM      1,628 bochsrc.bxrc
12/14/2008 12:02 PM      14,365 boot.asm
09/24/2020 11:11 PM        512 boot.bin
09/16/2015 07:51 AM        512 boots.bin
12/15/2008 12:47 AM         78 dosfp.bat
09/24/2020 11:17 PM    1,474,560 floppy.img
12/14/2008 11:45 AM      7,966 kernel.asm
12/15/2008 04:21 PM       227 Makefile
12/15/2008 12:20 PM        44 s.bat
      10 File(s)      1,510,732 bytes
      2 Dir(s)  43,603,685,376 bytes free

c:\OS\LAB\LAB1>del floppy.img

c:\OS\LAB\LAB1>make fp.disk
nasm boot.asm -o boot.bin -f bin
dd if=boot.bin of=floppy.img
rawwrite dd for windows version 0.5.
Written by John Newbigin <jn@it.swin.edu.au>
This program is covered by the GPL.  See copying.txt for details
1+0 records in
1+0 records out

c:\OS\LAB\LAB1>bximage
=====
                bximage
          Disk Image Creation Tool for Bochs
      $Id: bximage.c,v 1.32 2006/06/16 07:29:33 vruppert Exp $
=====

Do you want to create a floppy disk image or a hard disk image?
Please type hd or fd. [hd] fd

Choose the size of floppy disk image to create, in megabytes.
Please type 0.16, 0.18, 0.32, 0.36, 0.72, 1.2, 1.44, 1.68, 1.72, or 2.88.
[1.44] 1.44
I will create a floppy image with
  cyl=80
  heads=2
  sectors per track=18
  total sectors=2880
  total bytes=1474560

What should I name the image?
[a.img] _
```

## 12. Ketik 'floppya.img' untuk memberi nama image yang dibuat

```
Command Prompt - bximage
09/24/2020 11:11 PM        512 boot.bin
09/16/2015 07:51 AM        512 boots.bin
12/15/2008 12:47 AM         78 dosfp.bat
09/24/2020 11:17 PM    1,474,560 floppy.img
12/14/2008 11:45 AM      7,966 kernel.asm
12/15/2008 04:21 PM       227 Makefile
12/15/2008 12:20 PM        44 s.bat
      10 File(s)      1,510,732 bytes
      2 Dir(s)  43,603,685,376 bytes free

c:\OS\LAB\LAB1>del floppy.img

c:\OS\LAB\LAB1>make fp.disk
nasm boot.asm -o boot.bin -f bin
dd if=boot.bin of=floppy.img
rawwrite dd for windows version 0.5.
Written by John Newbigin <jn@it.swin.edu.au>
This program is covered by the GPL.  See copying.txt for details
1+0 records in
1+0 records out

c:\OS\LAB\LAB1>bximage
=====
                bximage
          Disk Image Creation Tool for Bochs
      $Id: bximage.c,v 1.32 2006/06/16 07:29:33 vruppert Exp $
=====

Do you want to create a floppy disk image or a hard disk image?
Please type hd or fd. [hd] fd

Choose the size of floppy disk image to create, in megabytes.
Please type 0.16, 0.18, 0.32, 0.36, 0.72, 1.2, 1.44, 1.68, 1.72, or 2.88.
[1.44] 1.44
I will create a floppy image with
  cyl=80
  heads=2
  sectors per track=18
  total sectors=2880
  total bytes=1474560

What should I name the image?
[a.img] floppya.img

The disk image 'floppya.img' already exists.  Are you sure you want to replace it?
Please type yes or no. [no] _
```

### 13. Ketik perintah 'yes' untuk me-replace floppy.img lama dengan yang baru

```
Command Prompt
c:\OS\LAB\LAB1>make fp.disk
nasm boot.asm -o boot.bin -f bin
dd if=boot.bin of=floppya.img
rawwrite dd for windows version 0.5.
Written by John Newbiggin <jn@it.swin.edu.au>
This program is covered by the GPL. See copying.txt for details
1+0 records in
1+0 records out

c:\OS\LAB\LAB1>bximage
=====
bximage
Disk Image Creation Tool for Bochs
$Id: bximage.c,v 1.32 2006/06/16 07:29:33 wruppert Exp $
=====

Do you want to create a floppy disk image or a hard disk image?
Please type hd or fd. [hd] fd

Choose the size of floppy disk image to create, in megabytes.
Please type 0.16, 0.18, 0.32, 0.36, 0.72, 1.2, 1.44, 1.68, 1.72, or 2.88.
[1.44] 1.44
I will create a floppy image with
  cyl=80
  heads=2
  sectors per track=18
  total sectors=2880
  total bytes=1474560

What should I name the image?
[a.img] floppya.img

The disk image 'floppya.img' already exists. Are you sure you want to replace it?
Please type yes or no. [no] yes

Writing: [ ] Done.

I wrote 1474560 bytes to floppya.img.

The following line should appear in your bochsrc:
  floppya: image="floppya.img", status=inserted
(The line is stored in your windows clipboard, use CTRL-V to paste)

Press any key to continue

c:\OS\LAB\LAB1>
```

### 14. Lakukan pemformatan pada file floppy.img dengan megetik perintah 'dosfp' untuk membuka konfigurasi PC-Simulator

```
Bochs for Windows - Console
$Id: bximage.c,v 1.32 2006/06/16 07:29:33 wruppert Exp $
=====

Do you want to create a floppy disk image or a hard disk image?
Please type fd

Choose the size of floppy disk image to create, in megabytes.
Please type 0.16, 0.18, 0.32, 0.36, 0.72, 1.2, 1.44, 1.68, 1.72, or 2.88.
[1.44] 1.44
I will create a floppy image with
  cyl=80
  heads=2
  sectors per track=18
  total sectors=2880
  total bytes=1474560

What should I name the image?
[a.img] fl

The disk image 'fl' already exists. Are you sure you want to replace it?
Please type yes or no. [no] yes

Writing: [ ] Done.

I wrote 1474560 bytes to fl.

The following line should appear in your bochsrc:
  floppya: image="fl", status=inserted
(The line is stored in your windows clipboard, use CTRL-V to paste)

Press any key to continue

c:\OS\LAB\LAB1>

Bochs for Windows - Display
=====
CD-ROM Device Driver for IDE (Four Channels Supported)
(C)Copyright Oak Technology Inc. 1993-1996
Driver Version : U340
Device Name : OSLAB
Transfer Mode : Programmed I/O
Drive 0: Port= 1F0 (Primary Channel), Slave IRQ= 14
Firmware version : ALPH

MSCDEX Version 2.23
Copyright (C) Microsoft Corp. 1986-1993. All rights reserved.
Drive B: = Driver OSLAB unit 0

a:\>_

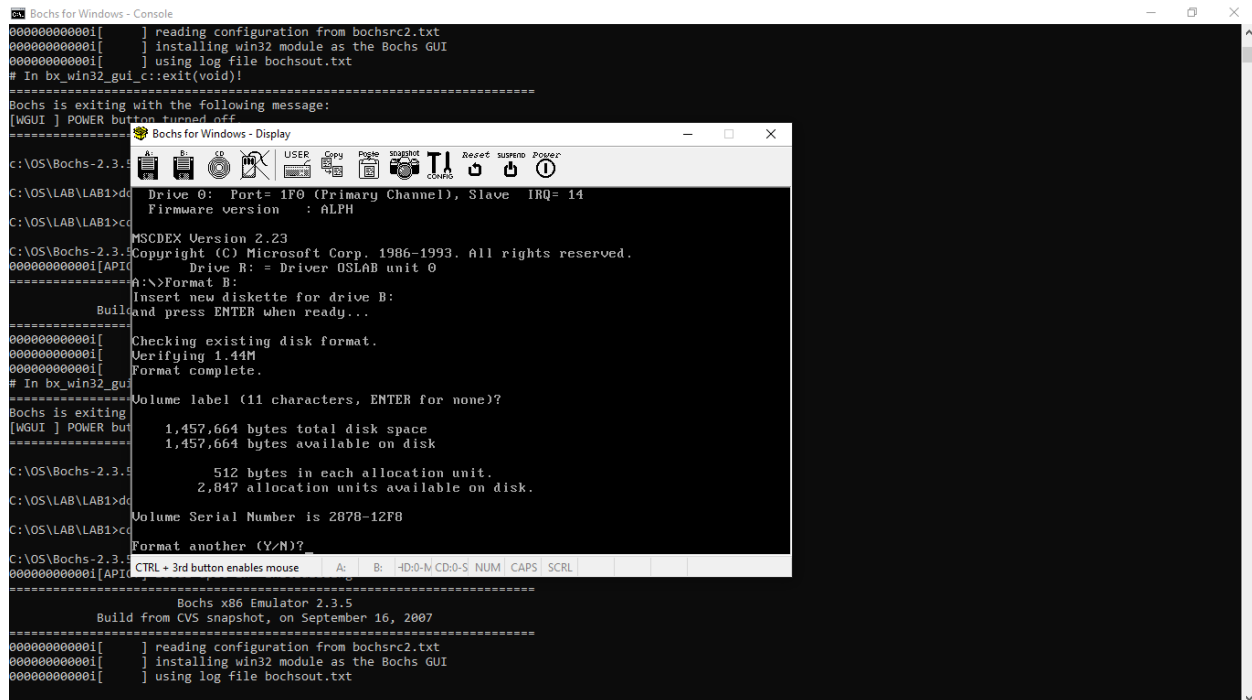
c:\OS\LAB\LAB1>

Bochs x86 Emulator 2.3.5
Build from CVS snapshot, on September 16, 2007

=====
0000000000i[ ] reading configuration from bochsrc2.txt
0000000000i[ ] installing win32 module as the Bochs GUI
0000000000i[ ] using log file bochsout.txt
```



## 15. Ketik 'Format B:' setelah format selesai klik enter



The screenshot shows two windows from the Bochs emulator. The 'Bochs for Windows - Console' window displays the following text:

```
00000000000i[ ] reading configuration from bochsrc2.txt
00000000000i[ ] installing win32 module as the Bochs GUI
00000000000i[ ] using log file bochsout.txt
# In bx_win32_gui.c::exit(void)!

Bochs is exiting with the following message:
[WGUI ] POWER button turned off.

=====
C:\OS\Bochs-2.3.5>
C:\OS\LAB\LAB1>dd if=0 of=1
Drive 0: Port= 1F0 (Primary Channel), Slave IRQ= 14
Firmware version : ALPH
C:\OS\LAB\LAB1>cd ..
C:\OS\Bochs-2.3.5>
MSCDEX Version 2.23
Copyright (C) Microsoft Corp. 1986-1993. All rights reserved.
Drive B: = Driver OSLAB unit 0
00000000000i[APIC?]
A:\>Format B:
Insert new diskette for drive B:
and press ENTER when ready...

=====
Checking existing disk format.
Verifying 1.44M
Format complete.
# In bx_win32_gui.c:
Volume label (11 characters, ENTER for none)?
Bochs is exiting
[WGUI ] POWER button
1,457,664 bytes total disk space
1,457,664 bytes available on disk

=====
512 bytes in each allocation unit.
2,847 allocation units available on disk.

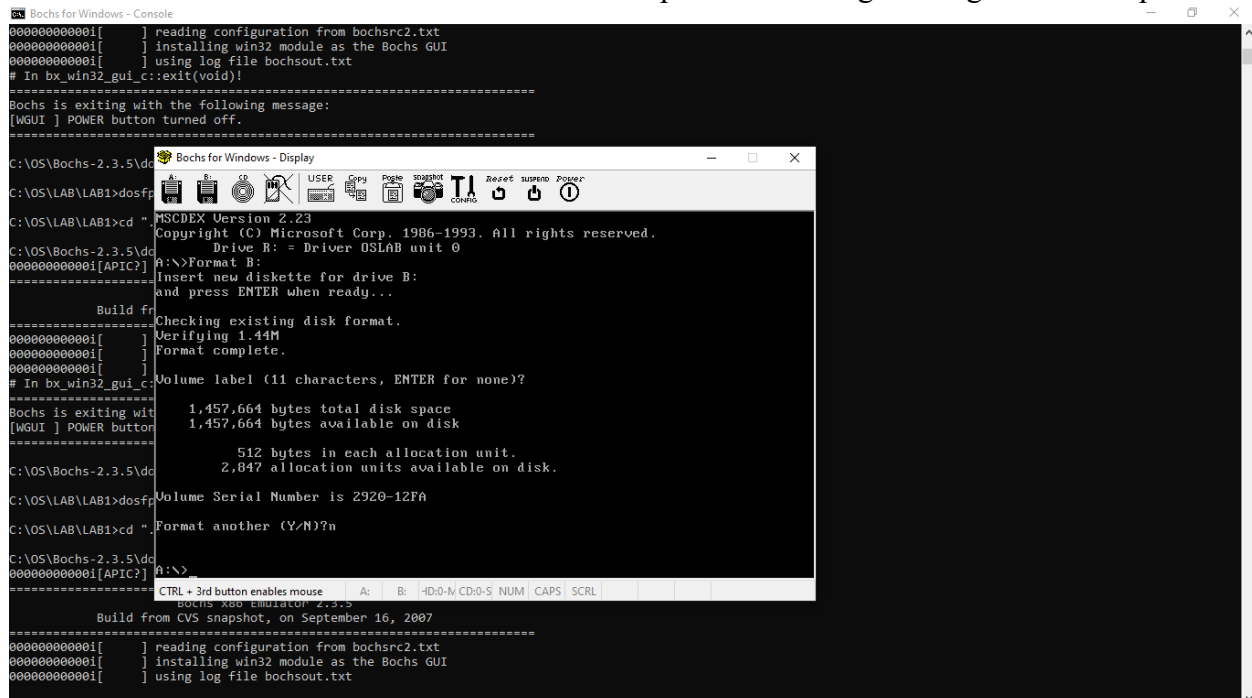
Volume Serial Number is 2878-12F8
C:\OS\LAB\LAB1>cd ..
Format another (Y/N)?
C:\OS\Bochs-2.3.5>
00000000000i[APIC?]
CTRL + 3rd button enables mouse
A: B: -HD:0-M:CD:0-S: NUM CAPS SCRL

=====
Bochs x86 Emulator 2.3.5
Build from CVS snapshot, on September 16, 2007

00000000000i[ ] reading configuration from bochsrc2.txt
00000000000i[ ] installing win32 module as the Bochs GUI
00000000000i[ ] using log file bochsout.txt
```

The 'Bochs for Windows - Display' window shows a graphical interface with a toolbar containing icons for USER, Copy, Paste, Snapshot, CTRL, Reset, Suspend, and Power. The main display area is black.

## 16. Karena sudah selesai kita ketik 'n' lalu tutup simulator dengan mengklik tombol power



The screenshot shows the same two windows as in the previous image. The 'Bochs for Windows - Console' window now shows the user typing 'n' in response to the 'Format another (Y/N)?' prompt:

```
Format another (Y/N)?n
C:\OS\Bochs-2.3.5>
00000000000i[APIC?]
CTRL + 3rd button enables mouse
A: B: -HD:0-M:CD:0-S: NUM CAPS SCRL

=====
Bochs x86 Emulator 2.3.5
Build from CVS snapshot, on September 16, 2007

00000000000i[ ] reading configuration from bochsrc2.txt
00000000000i[ ] installing win32 module as the Bochs GUI
00000000000i[ ] using log file bochsout.txt
```

The 'Bochs for Windows - Display' window remains the same, but the 'Power' button in the toolbar is now highlighted, indicating it has been clicked.

## 17. Memindah data dari file boots.bin ke memori kerja debug dengan perintah 'tdump boots.bin'

```
Command Prompt
Bochs is exiting with the following message:
[VGUI ] POWER button turned off.

C:\OS\Bochs-2.3.5>cd "C:\os\lab\lab1"

C:\OS\LAB\LAB1>tdump boots.bin
Turbo Dump Version 5.0.16.12 Copyright (c) 1988, 2000 Inprise Corporation
Display of File BOOTS.BIN

000000: EB 3C 90 40 53 57 49 4E 34 2E 31 00 02 01 01 00 .<.MSWIN4.1....
000010: 02 E0 00 40 0B F0 09 00 12 00 02 00 00 00 00 00 ...@.....
000020: 00 00 00 00 00 00 29 E7 0F 15 2C 4E 4F 20 4E 41 .....),NO NA
000030: 4D 45 20 20 20 20 46 41 54 31 32 20 20 20 33 C9 ME FAT12 3.
000040: 8E D1 BC FC 7B 16 07 8D 78 00 C5 76 00 1E 56 16 ....{...x.v..V.
000050: 55 BF 22 05 89 7E 00 89 4E 02 B1 0B FC F3 A4 06 U"...N.....
000060: 1F BD 00 7C C6 45 FE 0F 38 4E 24 7D 20 88 C1 99 ...|.E..BNS) ...
000070: E8 7E 01 83 EB 3A 66 A1 1C 7C 66 3B 07 8A 57 FC .....f..|fj..M.
000080: 75 06 80 CA 02 88 56 02 80 C3 10 73 ED 33 C9 FE u....V....s.3..
000090: 06 D8 7D 8A 46 10 98 F7 66 16 03 46 1C 13 56 1E ..).F...F..F..V.
0000A0: 03 46 0E 13 D1 8B 76 11 60 89 46 FC 89 56 FE B8 .F....V..^..F..V..
0000B0: 20 00 F7 E6 8B 5E 0B 03 C3 48 F7 F3 01 46 FC 11 ....^...H..F..
0000C0: 4E FE 61 BF 00 07 E8 28 01 72 3E 38 2D 74 17 60 N.a....(r>8-t..
0000D0: B1 0B BE D8 7D F3 A6 61 74 3D 4E 74 09 83 C7 20 ....).at-Nt....
0000E0: 3B FB 72 E7 EB DD FE 0E D8 7D 7B A7 BE 7F 7D AC ;.r.....){...}.
0000F0: 98 03 F0 AC 98 40 74 0C 48 74 13 B4 0E BB 07 00 ....@t.Ht.....
000100: CD 10 EB EF BE 02 7D EB E6 BE 80 7D EB E1 CD 16 .....}.....
000110: 5E 1F 66 8F 04 CD 19 BE 81 7D 8B 7D 1A 8D 45 FE ^..f.....}.E.
000120: 8A 4E 00 F7 E1 03 46 FC 13 56 FE B1 04 E8 C2 00 .N....F..V.....
000130: 72 D7 EA 00 02 70 00 52 50 06 53 6A 01 6A 10 91 r....p.RP.Sj.j..
000140: 8B 46 18 A2 26 05 96 92 33 D2 F7 F6 91 F7 F6 42 .F.&...3.....B
000150: 87 CA F7 76 1A 8A F2 8A E8 C0 CC 02 0A CC 88 01 ...v.....
000160: 02 80 7E 02 0E 75 04 B4 42 8B F4 8A 56 24 CD 13 ...u..B..V$.
000170: 61 61 72 0A 40 75 01 42 03 5E 08 49 75 77 C3 03 aa@.u.B.^..Iuw..
000180: 18 01 27 00 0A 49 6E 76 61 6C 69 64 20 73 79 73 ...Invalid sys
000190: 74 65 6D 20 64 69 73 68 FF 00 0A 44 69 73 68 20 tem disk...Disk
0001A0: 49 2F 4F 20 65 72 72 6F 72 FF 00 0A 52 65 70 6C I/O error...Repl
0001B0: 61 63 65 20 74 68 65 2B 64 69 73 68 2C 20 61 6E ace the disk, an
0001C0: 64 20 74 68 65 6E 20 70 72 65 73 73 20 61 6E 79 d then press any
0001D0: 20 68 65 79 00 0A 00 00 49 4F 20 20 20 20 20 20 key...IO
0001E0: 53 59 53 4D 53 44 4F 53 20 20 20 53 59 53 7F 01 SYSMSDOS SYS..
0001F0: 00 41 8B 00 07 60 66 6A 00 E9 3B FF 00 00 55 AA .A....^fj...U.

C:\OS\LAB\LAB1>
```

## 18. Masukkan perintah 's' untuk membuka simulator

```
Bochs for Windows - Console
000000: EB 3C 90 40 53 57 49 4E 34 2E 31 00 02 01 01 00 .<.MSWIN4.1....
000010: 02 E0 00 40 0B F0 09 00 12 00 02 00 00 00 00 00 ...@.....
000020: 00 00 00 00 00 00 29 E7 0F 15 2C 4E 4F 20 4E 41 .....),NO NA
000030: 4D 45 20 20 20 20 46 41 54 31 32 20 20 20 33 C9 ME FAT12 3.
000040: 8E D1 BC FC 7B 16 07 8D 78 00 C5 76 00 1E 56 16 ....{...x.v..V.
000050: 55 BF 22 05 89 7E 00 89 4E 02 B1 0B FC F3 A4 06 U"...N.....
000060: 1F BD 00 7C C6 45 FE 0F 38 4E 24 7D 20 88 C1 99 ...|.E..BNS) ...
000070: E8 7E 01 83 EB 3A 66 A1 1C 7C 66 3B 07 8A 57 FC .....f..|fj..M.
000080: 75 06 80 CA 02 88 56 02 80 C3 10 73 ED 33 C9 FE u....V....s.3..
000090: 06 D8 7D 8A 46 10 98 F7 66 16 03 46 1C 13 56 1E ..).F...F..F..V.
0000A0: 03 46 0E 13 D1 8B 76 11 60 89 46 FC 89 56 FE B8 .F....V..^..F..V..
0000B0: 20 00 F7 E6 8B 5E 0B 03 C3 48 F7 F3 01 46 FC 11 ....^...H..F..
0000C0: 4E FE 61 BF 00 07 E8 28 01 72 3E 38 2D 74 17 60 N.a....(r>8-t..
0000D0: B1 0B BE D8 7D F3 A6 61 74 3D 4E 74 09 83 C7 20 ....).at-Nt....
0000E0: 3B FB 72 E7 EB DD FE 0E D8 7D 7B A7 BE 7F 7D AC ;.r.....){...}.
0000F0: 98 03 F0 AC 98 40 74 0C 48 74 13 B4 0E BB 07 00 ....@t.Ht.....
000100: CD 10 EB EF BE 02 7D EB E6 BE 80 7D EB E1 CD 16 .....}.....
000110: 5E 1F 66 8F 04 CD 19 BE 81 7D 8B 7D 1A 8D 45 FE ^..f.....}.E.
000120: 8A 4E 00 F7 E1 03 46 FC 13 56 FE B1 04 E8 C2 00 .N....F..V.....
000130: 72 D7 EA 00 02 70 00 52 50 06 53 6A 01 6A 10 91 r....p.RP.Sj.j..
000140: 8B 46 18 A2 26 05 96 92 33 D2 F7 F6 91 F7 F6 42 .F.&...3.....B
000150: 87 CA F7 76 1A 8A F2 8A E8 C0 CC 02 0A CC 88 01 ...v.....
000160: 02 80 7E 02 0E 75 04 B4 42 8B F4 8A 56 24 CD 13 ...u..B..V$.
000170: 61 61 72 0A 40 75 01 42 03 5E 08 49 75 77 C3 03 aa@.u.B.^..Iuw..
000180: 18 01 27 00 0A 49 6E 76 61 6C 69 64 20 73 79 73 ...Invalid sys
000190: 74 65 6D 20 64 69 73 68 FF 00 0A 44 69 73 68 20 tem disk...Disk
0001A0: 49 2F 4F 20 65 72 72 6F 72 FF 00 0A 52 65 70 6C I/O error...Repl
0001B0: 61 63 65 20 74 68 65 2B 64 69 73 68 2C 20 61 6E ace the disk, an
0001C0: 64 20 74 68 65 6E 20 70 72 65 73 73 20 61 6E 79 d then press any
0001D0: 20 68 65 79 00 0A 00 00 49 4F 20 20 20 20 20 20 key...IO
0001E0: 53 59 53 4D 53 44 4F 53 20 20 20 53 59 53 7F 01 SYSMSDOS SYS..
0001F0: 00 41 8B 00 07 60 66 6A 00 E9 3B FF 00 00 55 AA .A....^fj...U.

C:\OS\LAB\LAB1>s
C:\OS\LAB\LAB1>.\..\bo
00000000000i[APIC?] loc
=====
Build from CTRL + 3rd button enables mouse A: NUM CAPS SCRL
00000000000i[ ] reading configuration from bochsrc.bxrc
00000000000i[ ] installing win32 module as the Bochs GUI
00000000000i[ ] using log file bochsout.txt
```

19. Boot invalid karena floppy.img belum diisi system file, langkah berikutnya yaitu mengetik perintah ‘del floppy.img’, membuat img baru dengan perintah ‘bximage’ lalu pilih type fd dan ukuran 1.44 MB. Lalu dinamai dengan floppy.img

```
Command Prompt
=====
00000000000i[      ] reading configuration from bochsrc.bxrc
00000000000i[      ] installing win32 module as the Bochs GUI
00000000000i[      ] using log file bochsout.txt
# In bx_win32_gui_c::exit(void)!
=====
Bochs is exiting with the following message:
[VGUI ] POWER button turned off.
=====

C:\OS\LAB\LAB1>del floppy.img

C:\OS\LAB\LAB1>bximage
=====
                bximage
        Disk Image Creation Tool for Bochs
        $Id: bximage.c,v 1.32 2006/06/16 07:29:33 vruppert Exp $
=====

Do you want to create a floppy disk image or a hard disk image?
Please type hd or fd. [hd] fd

Choose the size of floppy disk image to create, in megabytes.
Please type 0.16, 0.18, 0.32, 0.36, 0.72, 1.2, 1.44, 1.68, 1.72, or 2.88.
[1.44] 1.44
I will create a floppy image with
  cyl=80
  heads=2
  sectors per track=18
  total sectors=2880
  total bytes=1474560

What should I name the image?
[a.img] floppy.img

Writing: [ ] Done.

I wrote 1474560 bytes to floppy.img.

The following line should appear in your bochsrc:
  floppy: image="floppy.img", status=inserted
(The line is stored in your windows clipboard, use CTRL-V to paste)

Press any key to continue

C:\OS\LAB\LAB1>
```

20. Buka simulator dengan ketik ‘dosfp’ lalu masukkan perintah ‘Format B:/S’, lalu enter untuk lanjut

```
Bochs for Windows - Console
=====
Bochs for Windows - Display
=====
MSCDEX Version 2.23
Copyright (C) Microsoft Corp. 1986-1993. All rights reserved.
Drive R: = Driver OS/LAB unit 0

A:\>Format B:/S
Insert new diskette for drive B:
and press ENTER when ready...

Checking existing disk format.
Formatting 1.44M
Format complete.
System transferred

Volume label (11 characters, ENTER for none)?

1,457,664 bytes total disk space
221,696 bytes used by system
1,235,968 bytes available on disk

512 bytes in each allocation unit.
2,414 allocation units available on disk.

Volume Serial Number is 435A-1314

Format another (Y/N)?

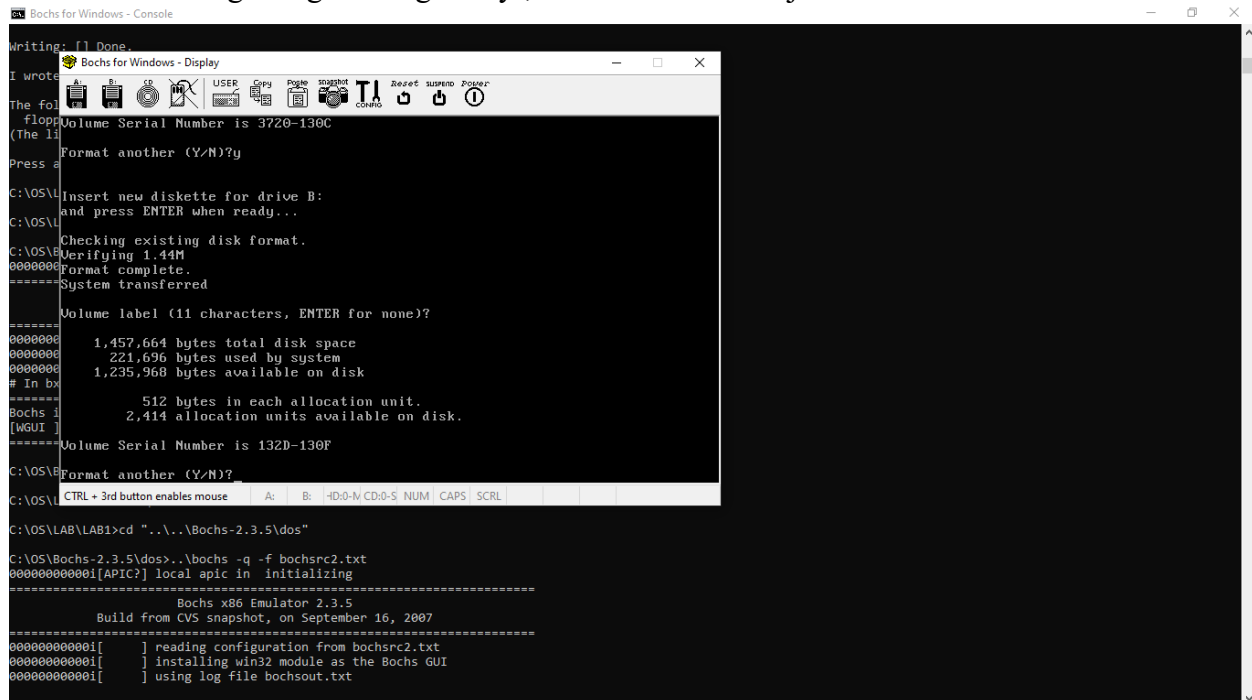
Press CTRL + 3rd button enables mouse  A: B: -HD-D-M CD-D-S NUM | CAPS | SCRL |

C:\OS\LAB\LAB1>dosfp

C:\OS\LAB\LAB1>cd "..\..\Bochs-2.3.5\dos"

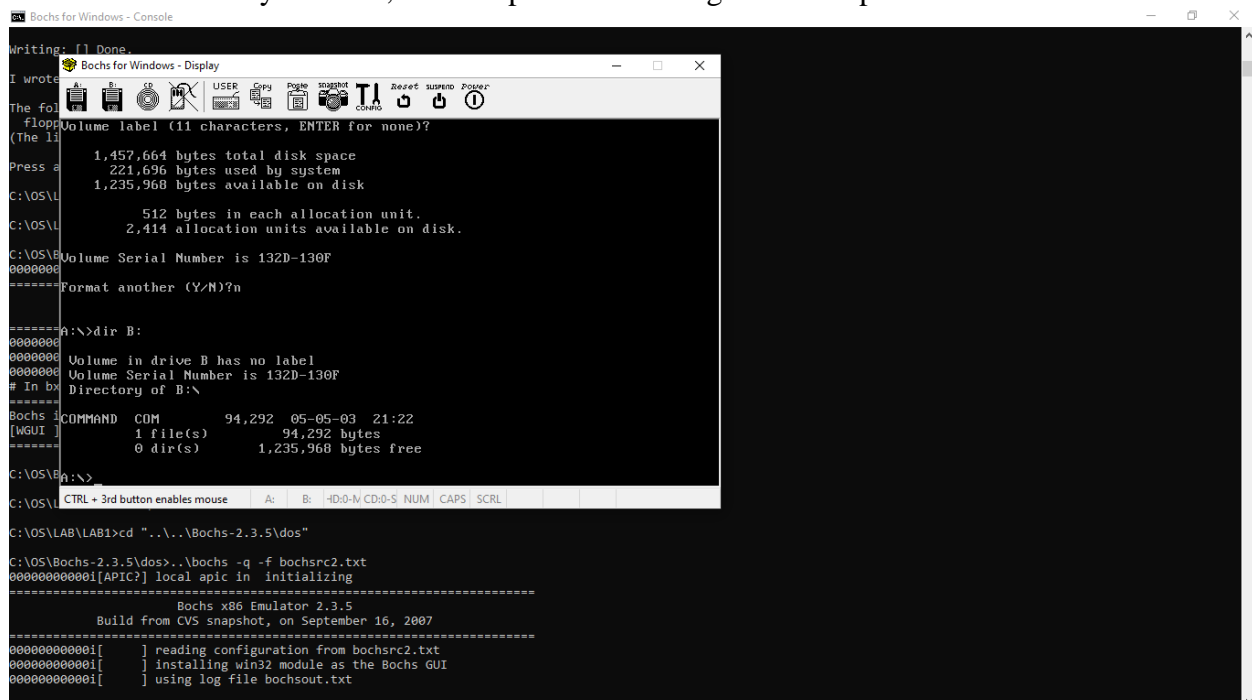
C:\OS\Bochs-2.3.5>dos>..\bochs -q -f bochsrc2.txt
00000000000i[APIC?] local apic in initializing
=====
                Bochs x86 Emulator 2.3.5
        Build from CVS snapshot, on September 16, 2007
=====
00000000000i[      ] reading configuration from bochsrc2.txt
00000000000i[      ] installing win32 module as the Bochs GUI
00000000000i[      ] using log file bochsout.txt
```

## 21. Format lagi dengan mengetik 'y', lalu enter untuk lanjut



```
Bochs for Windows - Console
Writing: [ ] Done
I wrote
The floppy disk image
Volume Serial Number is 3720-130C
Format another (Y/N)?y
Press a
C:\OS\LAB\LAB1>Insert new diskette for drive B:
and press ENTER when ready...
C:\OS\LAB\LAB1>Checking existing disk format.
C:\OS\LAB\LAB1>Verifying 1.44M
00000000
Format complete.
=====
System transferred
Volume label (11 characters, ENTER for none)?
00000000 1,457,664 bytes total disk space
00000000 221,696 bytes used by system
00000000 1,235,968 bytes available on disk
# In by
=====
512 bytes in each allocation unit.
2,414 allocation units available on disk.
Bochs [
[WGUI ]
=====
Volume Serial Number is 132D-130F
C:\OS\LAB\LAB1>Format another (Y/N)?n
C:\OS\LAB\LAB1>CTRL + 3rd button enables mouse A: B: <D> <M> <CD> <0> <S> NUM CAPS SCRL
C:\OS\LAB\LAB1>cd "..\..\Bochs-2.3.5\dos"
C:\OS\Bochs-2.3.5\dos>.\bochs -q -f bochsrc2.txt
00000000000i[APIC?] local apic in initializing
=====
Bochs x86 Emulator 2.3.5
Build from CVS snapshot, on September 16, 2007
00000000000i[ ] reading configuration from bochsrc2.txt
00000000000i[ ] installing win32 module as the Bochs GUI
00000000000i[ ] using log file bochsout.txt
```

## 22. Ketik 'n' karena format selesai, lalu ketik 'dir B:' untuk mengecek bahwa floppy.img telah terisi system file, lalu tutup simulator dengan tombol power



```
Bochs for Windows - Console
Writing: [ ] Done
I wrote
The floppy disk image
Volume label (11 characters, ENTER for none)?
1,457,664 bytes total disk space
221,696 bytes used by system
1,235,968 bytes available on disk
512 bytes in each allocation unit.
2,414 allocation units available on disk.
Volume Serial Number is 132D-130F
Format another (Y/N)?n
A:\>dir B:
00000000
00000000 Volume in drive B has no label
00000000 Volume Serial Number is 132D-130F
# In by
Directory of B:\
=====
Bochs [COMMAND COM 94,292 05-05-03 21:22
[WGUI ] 1 file(s) 94,292 bytes
===== 0 dir(s) 1,235,968 bytes free
C:\OS\LAB\LAB1>A:\>
C:\OS\LAB\LAB1>CTRL + 3rd button enables mouse A: B: <D> <M> <CD> <0> <S> NUM CAPS SCRL
C:\OS\LAB\LAB1>cd "..\..\Bochs-2.3.5\dos"
C:\OS\Bochs-2.3.5\dos>.\bochs -q -f bochsrc2.txt
00000000000i[APIC?] local apic in initializing
=====
Bochs x86 Emulator 2.3.5
Build from CVS snapshot, on September 16, 2007
00000000000i[ ] reading configuration from bochsrc2.txt
00000000000i[ ] installing win32 module as the Bochs GUI
00000000000i[ ] using log file bochsout.txt
```

Bochs for Windows - Console

Build from CVS snapshot, on September 16, 2007

```

=====
00000000000i[      ] reading configuration from bochsrc2.txt
00000000000i[      ] installing win32 module as the Bochs GUI
00000000000i[      ] using log file bochsout.txt
=====

```

Bochs for Windows - Display

USER

Bochs is e

[WGUI ] PO

=====

Plex86/Bochs UGABios 0.6a 19 Aug 2006

This UGA/UEFI Bios is released under the GNU LGPL

C:\OS\Boch

Please visit :

- . <http://bochs.sourceforge.net>
- . <http://www.nongnu.org/vgabios>

C:\OS\LAB

Bochs VBE Display Adapter enabled

C:\OS\Boch

Bochs BIOS - build: 09/10/07

00000000000i[ ] Revision: 1.183 \$ \$Date: 2007/09/10 20:00:29 \$

Options: apmbios pcbios eltorito rombios32

=====

Booting from Floppy...

Starting BootCD.....

00000000000i[ ]

00000000000i[ ]

00000000000i[ ]

# In bx wi

Microsoft(R) MS-DOS 7.1

(C)Copyright Microsoft Corp 1981-1999.

[WGUI ] PO

A:\>\_

C:\OS\Boch

CTRL + 3rd button enables mouse

A: NUM CAPS SCRL

C:\OS\LAB

C:\OS\LAB\LAB1>..\bochs-2.3.5\bochs -q -f bochsrc.bxrc

00000000000i[APIC?] local apic in initializing

=====

Bochs x86 Emulator 2.3.5

Build from CVS snapshot, on September 16, 2007

```

=====
00000000000i[      ] reading configuration from bochsrc.bxrc
00000000000i[      ] installing win32 module as the Bochs GUI
00000000000i[      ] using log file bochsout.txt
=====

```

1. Apa yang dimaksud dengan kode 'ASCII', buatlah tabel kode ASCII lengkap cukup kode

Jawab :

1. **ASCII** (*American Standard Code for Information Interchange*) merupakan kode standar Amerika untuk pertukaran informasi atau sebuah standar internasional dalam pengkodean huruf dan simbol seperti Unicode dan Hex, tetapi ASCII lebih bersifat universal.

**Tabel ASCII**

Dec	Bin	Hex	Char
32	010 0000	20	[space]
33	010 0001	21	!
34	010 0010	22	"
35	010 0011	23	#
36	010 0100	24	\$
37	010 0101	25	%
38	010 0110	26	&
39	010 0111	27	'
40	010 1000	28	(
41	010 1001	29	)
42	010 1010	2A	*
43	010 1011	2B	+
44	010 1100	2C	,
45	010 1101	2D	-
46	010 1110	2E	.
47	010 1111	2F	/
48	011 0000	30	0
49	011 0001	31	1
50	011 0010	32	2
51	011 0011	33	3
52	011 0100	34	4
53	011 0101	35	5
54	011 0110	36	6
55	011 0111	37	7
56	011 1000	38	8
57	011 1001	39	9
58	011 1010	3A	:
59	011 1011	3B	;
60	011 1100	3C	<
61	011 1101	3D	=
62	011 1110	3E	>
63	011 1111	3F	?

Dec	Bin	Hex	Char
64	100 0000	40	@
65	100 0001	41	A
66	100 0010	42	B
67	100 0011	43	C
68	100 0100	44	D
69	100 0101	45	E
70	100 0110	46	F
71	100 0111	47	G
72	100 1000	48	H
73	100 1001	49	I
74	100 1010	4A	J
75	100 1011	4B	K
76	100 1100	4C	L
77	100 1101	4D	M
78	100 1110	4E	N
79	100 1111	4F	O
80	101 0000	50	P
81	101 0001	51	Q
82	101 0010	52	R
83	101 0011	53	S
84	101 0100	54	T
85	101 0101	55	U
86	101 0110	56	V
87	101 0111	57	W
88	101 1000	58	X
89	101 1001	59	Y
90	101 1010	5A	Z
91	101 1011	5B	[
92	101 1100	5C	\
93	101 1101	5D	]
94	101 1110	5E	^
95	101 1111	5F	_

Dec	Bin	Hex	Char
96	110 0000	60	`
97	110 0001	61	a
98	110 0010	62	b
99	110 0011	63	c
100	110 0100	64	d
101	110 0101	65	e
102	110 0110	66	f
103	110 0111	67	g
104	110 1000	68	h
105	110 1001	69	i
106	110 1010	6A	j
107	110 1011	6B	k
108	110 1100	6C	l
109	110 1101	6D	m
110	110 1110	6E	n
111	110 1111	6F	o
112	111 0000	70	p
113	111 0001	71	q
114	111 0010	72	r
115	111 0011	73	s
116	111 0100	74	t
117	111 0101	75	u
118	111 0110	76	v
119	111 0111	77	w
120	111 1000	78	x
121	111 1001	79	y
122	111 1010	7A	z
123	111 1011	7B	{
124	111 1100	7C	
125	111 1101	7D	}
126	111 1110	7E	~
127	111 1111	7F	[del]

## 2. Bahasa Assembly untuk x86, terbagi menjadi 3 bagian utama yaitu :

### 1. Komentar

Komentar diawali dengan tanda titik koma (;).

; ini adalah komentar

### 2. Label

Label diakhiri dengan tanda titik dua (:).

Contoh: main: ,loop: ,proses: ,keluar:

### 3. Assembler directives

Directives adalah perintah yang ditujukan kepada assembler ketika sedang menerjemahkan program kita ke bahasa mesin.

Directive dimulai dengan tanda titik. **.model** : memberitahu assembler berapa memori yang akan dipakai oleh program kita.

Ada model tiny, model small, model compact, model medium, model large, dan model huge.

**.data** : memberitahu assembler bahwa bagian di bawah ini adalah data program.

**.code** : memberitahu assembler bahwa bagian di bawah ini adalah instruksi program.

**.stack** : memberitahu assembler bahwa program kita memiliki stack.

Program EXE harus punya stack. Kira-kira yang penting itu dulu.

Semua directive yang dikenal assembler adalah: .186 .286 .286c .286p .287 .386 .386c .386p .387 .486 .486p .8086 .8087 .alpha .break .code .const .continue .cref .data .data? .dosseg .else .elseif .endif .endw .err .err1 .err2 .errb .errdef .errdif .errdifi .erre .erridn .erridni .errnb .errndef .errnz .exit .fardata .fardata? .if .lall .lfcond .list .listall .listif .listmacro .listmacroall .model .no87 .nocref .nolist .nolistif .nolistmacro .radix .repeat .sall .seq .sfcond .stack .startup .tfcond .type .until .untilcxz .while .xall .xcref .xlist.

## Definisi data

**DB** : define bytes. Membentuk data byte demi byte. Data bisa data numerik maupun teks.

catatan: untuk membentuk data string, pada akhir string harus diakhiri tanda dolar (\$).

sintaks: {label} DB {data} contoh: teks1 db "Hello world \$"

**DW** : define words.

Membentuk data word demi word (1 word = 2 byte).



sintaks: {label} DW {data} contoh: kucing dw ?, ?, ? ;mendefinisikan tiga slot 16-bit yang isinya don't care

(disimbolkan dengan tanda tanya)

**DD** : define double words. Membentuk data doubleword demi doubleword (4 byte).

sintaks: {label} DD {data}

**EQU** : equals. Membentuk konstanta. sintaks: {label} EQU {data}

contoh: sepuluh EQU 10

Ada assembly yang melibatkan bilangan pecahan (floating point), bilangan bulat (integer), DF (define far words),

DQ (define quad words), dan DT (define ten bytes).

### **Perpindahan data**

**MOV** : move. Memindahkan suatu nilai dari register ke memori, memori ke register, atau register ke register.

sintaks: MOV {tujuan}, {sumber}

contoh:

*mov AX, 4C00h ;mengisi register AX dengan 4C00(hex).*

*mov BX, AX ;menyalin isi AX ke BX. mov CL, [BX] ;mengisi register CL dengan data di memori yang alamatnya ditunjuk BX.*

*mov CL, [BX] + 2 ;mengisi CL dengan data di memori yang alamatnya ditunjuk BX lalu geser maju 2 byte.*

*mov [BX], AX ;menyimpan nilai AX pada tempat di memori yang ditunjuk BX. mov [BX] - 1, 00101110b*

*;menyimpan 00101110(bin) pada alamat yang ditunjuk BX lalu geser mundur 1 byte.*

**LEA** : load effective address. Mengisi suatu register dengan alamat offset sebuah data.

sintaks: LEA {register}, {sumber} contoh: lea DX, teks1

**XCHG** : exchange. Menukar dua buah register langsung.

sintaks: XCHG {register 1}, {register 2} Kedua register harus punya ukuran yang sama.

Bila sama-sama 8 bit (misalnya AH dengan BL) atau sama-sama 16 bit (misalnya CX dan DX), maka pertukaran bisa dilakukan. Sebenarnya masih banyak perintah perpindahan data, misalnya

IN, OUT, LODS, LODSB, LODSW, MOVS, MOVSB, MOVSW, LDS, LES, LAHF, SAHF, dan XLAT.

### Operasi logika

**AND** : melakukan bitwise and. sintaks: AND {register}, {angka} AND {register 1}, {register 2} hasil disimpan di register 1.

contoh: mov AL, 00001011b mov AH, 11001000b and AL, AH ;sekarang AL berisi 00001000(bin), sedangkan AH tidak berubah.

**OR** : melakukan bitwise or. sintaks: OR {register}, {angka} OR {register 1}, {register 2} hasil disimpan di register 1.

**NOT** : melakukan bitwise not (*one's complement*) sintaks: NOT {register} hasil disimpan di register itu sendiri.

**XOR** : melakukan bitwise eksklusif or. sintaks: XOR {register}, {angka} XOR {register 1}, {register 2} hasil disimpan di register 1. Tips: sebuah register yang di-XOR-kan dengan dirinya sendiri akan menjadi berisi nol.

**SHL** : shift left. Menggeser bit ke kiri. Bit paling kanan diisi nol. sintaks: SHL {register}, {banyaknya}

**SHR** : shift right. Menggeser bit ke kanan. Bit paling kiri diisi nol. sintaks: SHR {register}, {banyaknya}

**ROL** : rotate left. Memutar bit ke kiri. Bit paling kiri jadi paling kanan kali ini. sintaks: ROL {register}, {banyaknya} Bila banyaknya rotasi tidak disebutkan, maka nilai yang ada di CL akan digunakan sebagai banyaknya rotasi.

**ROR** : rotate right. Memutar bit ke kanan. Bit paling kanan jadi paling kiri. sintaks: ROR {register}, {banyaknya} Bila banyaknya rotasi tidak disebutkan, maka nilai yang ada di CL akan digunakan sebagai banyaknya rotasi.

Ada lagi : RCL dan RCR.

### Operasi matematika

**ADD** : add. Menjumlahkan dua buah register.

sintaks: ADD {tujuan}, {sumber} operasi yang terjadi: tujuan = tujuan + sumber. carry (bila ada) disimpan di CF.

**ADC** : add with carry. Menjumlahkan dua register dan carry flag (CF).

sintaks: ADC {tujuan}, {sumber} operasi yang terjadi:  $\text{tujuan} = \text{tujuan} + \text{sumber} + \text{CF}$ . carry (bila ada lagi) disimpan lagi di CF.

**INC** : increment. Menjumlah isi sebuah register dengan 1.

Bedanya dengan ADD, perintah INC hanya memakan 1 byte memori sedangkan ADD pakai 3 byte.

sintaks: INC {register}

**SUB** : subtract. Mengurangkan dua buah register.

sintaks: SUB {tujuan}, {sumber} operasi yang terjadi:  $\text{tujuan} = \text{tujuan} - \text{sumber}$ . borrow (bila terjadi) menyebabkan CF bernilai 1.

**SBB** : subtract with borrow. Mengurangkan dua register dan carry flag (CF).

sintaks: SBB {tujuan}, {sumber} operasi yang terjadi:  $\text{tujuan} = \text{tujuan} - \text{sumber} - \text{CF}$ . borrow (bila terjadi lagi) menyebabkan CF dan SF (sign flag) bernilai 1.

**DEC** : decrement. Mengurang isi sebuah register dengan 1.

Jika SUB memakai 3 byte memori, DEC hanya memakai 1 byte. sintaks: DEC {register}

**MUL** : multiply. Mengalikan register dengan AX atau AH.

sintaks: MUL {sumber} Bila register sumber adalah 8 bit, maka isi register itu dikali dengan isi AL, kemudian disimpan di AX. Bila register sumber adalah 16 bit, maka isi register itu dikali dengan isi AX, kemudian hasilnya disimpan di DX:AX. Maksudnya, DX berisi high order byte-nya, AX berisi low order byte-nya.

**IMUL** : signed multiply. Sama dengan MUL, hanya saja IMUL menganggap bit-bit yang ada di register sumber sudah dalam bentuk *two's complement*.

sintaks: IMUL {sumber}

**DIV** : divide. Membagi AX atau DX:AX dengan sebuah register.

sintaks: DIV {sumber} Bila register sumber adalah 8 bit (misalnya: BL), maka operasi yang terjadi: AX dibagi BL, hasil bagi disimpan di AL, sisa bagi disimpan di AH. Bila register sumber adalah 16 bit (misalnya: CX), maka operasi yang terjadi: DX:AX dibagi CX, hasil bagi disimpan di AX, sisa bagi disimpan di DX.

**IDIV** : signed divide. Sama dengan DIV, hanya saja IDIV menganggap bit-bit yang ada di register sumber sudah dalam bentuk *two's complement*.

sintaks: IDIV {sumber}

**NEG** : negate. Membuat isi register menjadi negatif (*two's complement*). Bila mau *one's complement*, gunakan perintah NOT. sintaks: NEG {register} hasil disimpan di register itu sendiri.

### **Pengulangan**

**LOOP** : loop. Mengulang sebuah proses. Pertama register CX dikurangi satu. Bila CX sama dengan nol, maka looping berhenti. Bila tidak nol, maka lompat ke label tujuan.

sintaks: LOOP {label tujuan} Tips: isi CX dengan nol untuk mendapat jumlah pengulangan terbanyak. karena nol dikurang satu sama dengan -1, atau dalam notasi *two's complement* menjadi FFFF(hex) yang sama dengan 65535(dec).

**LOOPE** : loop while equal. Melakukan pengulangan selama  $CX \neq 0$  dan  $ZF = 1$ . CX tetap dikurangi 1 sebelum diperiksa.

sintaks: LOOP {label tujuan}

**LOOPZ** : loop while zero. Identik dengan LOOPE.

**LOOPNE** : loop while not equal.

Melakukan pengulangan selama  $CX \neq 0$  dan  $ZF = 0$ . CX tetap dikurangi 1 sebelum diperiksa.

sintaks: LOOPNE {label tujuan}

**LOOPNZ** : loop while not zero. Identik dengan LOOPNE.

**REP** : repeat. Mengulang perintah sebanyak CX kali. sintaks: REP {perintah assembly} contoh: *mov CX, 05 rep inc BX ;register BX ditambah 1 sebanyak 5x.*

**REPE** : repeat while equal. Mengulang perintah sebanyak CX kali, tetapi pengulangan segera dihentikan bila didapati  $ZF = 1$ .

sintaks: REPE {perintah assembly}

**REPZ** : repeat while zero. Identik dengan REPE.

**REPNE** : repeat while not equal. Mengulang perintah sebanyak CX kali, tetapi pengulangan segera dihentikan bila didapati  $ZF = 0$ .

sintaks: REPNE {perintah assembly}

**REPNZ** : repeat while not zero. Identik dengan REPNE.

### **Perbandingan**

**CMP** : compare. Membandingkan dua buah operand. Hasilnya mempengaruhi sejumlah flag register.

sintaks: `CMP {operand 1}, {operand 2}`. Operand ini bisa register dengan register, register dengan isi memori, atau register dengan angka. `CMP` tidak bisa membandingkan isi memori dengan isi memori. Hasilnya adalah:

Kasus	Bila operand 1 < operand 2	Bila operand 1 = operand 2	Bila operand 1 > operand 2
Signed binary	OF = 1, SF = 1, ZF = 0	OF = 0, SF = 0, ZF = 1	OF = 0, SF = 0, ZF = 0
Unsigned binary	CF = 1, ZF = 0	CF = 0, ZF = 1	CF = 0, ZF = 0

### Lompat-lompat

**JMP**: jump. Lompat tanpa syarat. Lompat begitu saja. sintaks: `JMP {label tujuan}`

**Lompat bersyarat** sintaksnya sama dengan `JMP`, yaitu perintah jump diikuti label tujuan.

PERINTAH	ARTI	SYARAT	KASUS	KETERANGAN ("OP" = OPERAND)	MENGIKUTI CMP?
<b>JA</b>	jump if above	$CF = 0 \wedge ZF = 0$	unsigned	lompat bila $op\ 1 > op\ 2$	ya
<b>JNBE</b>	jump if not below or equal				
<b>JB</b>	jump if below	$CF = 1 \wedge ZF = 0$	unsigned	lompat bila $op\ 1 < op\ 2$	ya
<b>JNAE</b>	jump if not above or equal				
<b>JAE</b>	jump if above or equal	$CF = 0 \vee ZF = 1$	unsigned	lompat bila $op\ 1 \geq op\ 2$	ya
<b>JNB</b>	jump if not below				
<b>JBE</b>	jump if below or equal	$CF = 1 \vee ZF = 1$	unsigned	lompat bila $op\ 1 \leq op\ 2$	ya
<b>JNA</b>	jump if not above				
<b>JG</b>	jump if greater	$OF = 0 \wedge ZF = 0$	signed	lompat bila $op\ 1 > op\ 2$	ya
<b>JNLE</b>	jump if not less or equal				

<b>JGE</b>	jump if greater or equal	$OF = 0 \vee ZF = 1$	signed	lompat bila $op\ 1 \geq op\ 2$	ya
<b>JNL</b>	jump if not less than				
<b>JL</b>	jump if less than	$OF = 1 \wedge ZF = 0$	signed	lompat bila $op\ 1 < op\ 2$	ya
<b>JNGE</b>	jump if not greater or equal				
<b>JLE</b>	jump if less or equal	$OF = 1 \vee ZF = 1$	signed	lompat bila $op\ 1 \leq op\ 2$	ya
<b>JNG</b>	jump if not greater				
<b>JE</b>	jump if equal	$ZF = 1$	keduanya	lompat bila $op\ 1 = op\ 2$	ya
<b>JZ</b>	jump if zero	$ZF = 1$	keduanya	lompat bila $op\ 1 = op\ 2$	ya
<b>JNE</b>	jump if not equal	$ZF = 0$	keduanya	lompat bila $op\ 1 \neq op\ 2$	ya
<b>JNZ</b>	jump if not zero	$ZF = 0$	keduanya	lompat bila $op\ 1 \neq op\ 2$	ya
<b>JC</b>	jump if carry	$CF = 1$	N/A	lompat bila carry flag = 1	tidak
<b>JNC</b>	jump if not carry	$CF = 0$	N/A	lompat bila carry flag = 0	tidak
<b>JP</b>	jump on parity	$PF = 1$	N/A	lompat bila parity flag = 1	tidak selalu
<b>JPE</b>	jump on parity even			lompat bila bilangan genap	
<b>JNP</b>	jump on not parity	$PF = 0$	N/A	lompat bila parity flag = 0	tidak selalu
<b>JPO</b>	jump on parity odd			lompat bila bilangan ganjil	
<b>JO</b>	jump if overflow	$OF = 1$	N/A	lompat bila overflow flag = 1	tidak

<b>JNO</b>	jump if not overflow	OF = 0	N/A	lompat bila overflow flag = 0	tidak
<b>JS</b>	jump if sign	SF = 1	N/A	lompat bila bilangan negatif	tidak
<b>JCXZ</b>	jump if CX is zero	CX = 0000	N/A	lompat bila CX berisi nol	tidak

## Operasi stack

**PUSH** : push. Menambahkan sesuatu ke stack. Sesuatu ini harus register berukuran 16 bit (pada 386+ harus 32 bit), tidak boleh angka, tidak boleh alamat memori. Maka Anda tidak bisa mem-push register 8-bit seperti AH, AL, BH, BL, dan kawan-kawannya.

sintaks: push {register 16-bit sumber}

contoh: push DX push AX Setelah operasi push, register SP (stack pointer) otomatis dikurangi 2 (karena datanya 2 byte). Makanya, “top” dari stack seakan-akan “tumbuh turun”.

**POP** : pop. Mengambil sesuatu dari stack. Sesuatu ini akan disimpan di register tujuan dan harus 16-bit. Maka Anda tidak bisa mem-pop menuju AH, AL, dkk.

sintaks: POP {register 16-bit tujuan}

contoh: POP BX Setelah operasi pop, register SP otomatis ditambah 2 (karena 2 byte), sehingga “top” dari stack “naik” lagi.

Tip: karena register segmen tidak bisa diisi langsung nilainya, Anda bisa menggunakan stack sebagai perantaranya.

Contoh kodenya: mov AX, seg teks1 push AX pop DS

**PUSHF** : push flags. Mem-push **semua** isi register flag ke dalam stack. Biasa dipakai untuk *membackup* data di register flag sebelum operasi matematika. Sintaks: PUSHF ;(saja).

**POPF** : pop flags. Lawan dari pushf. Sintaks: POPF ;(saja).

**POPA** : pop all general-purpose registers. Adalah ringkasan dari sejumlah perintah dengan urutan: *pop DI pop SI pop BP pop SP pop BX pop DX pop CX pop AX*. Urutan sudah ditetapkan seperti itu.

sintaks: POPA ;(saja). Jauh lebih cepat mengetikkan POPA daripada mengetik POP-POP-POP yang banyak itu.

**PUSHA** : push all general-purpose registers. Lawan dari POPA, dimana PUSHA adalah singkatan dari sejumlah perintah dengan urutan yang sudah ditetapkan: *push AX push CX push DX push BX push SP push BP push SI push DI*.

### Operasi pada register flag

**CLC** : clear carry flag. Menjadikan  $CF = 0$ . Sintaks: CLC ;(saja).

**STC** : set carry flag. Menjadikan  $CF = 1$ . Sintaks: STC ;(saja).

**CMC** : complement carry flag. Melakukan operasi NOT pada CF. Yang tadinya 0 menjadi 1, dan sebaliknya.

**CLD** : clear direction flag. Menjadikan  $DF = 0$ . Sintaks: CLD ;(saja).

**STD** : set direction flag. Menjadikan  $DF = 1$ .

**CLI** : clear interrupt flag. Menjadikan  $IF = 0$ , sehingga interrupt ke CPU akan di-disable. Biasanya perintah CLI diberikan sebelum menjalankan sebuah proses penting yang riskan gagal bila diganggu.

**STI** : set interrupt flag. Menjadikan  $IF = 1$ . Perintah lainnya

**ORG** : origin. Mengatur awal dari program (bagian static data). Analoginya seperti mengatur dimana letak titik (0, 0) pada koordinat Cartesius.

sintaks: ORG {alamat awal}

Pada program COM (program yang berekstensi .com), harus ditulis “ORG 100h” untuk mengatur alamat mulai dari program pada 0100(hex), karena dari alamat 0000(hex) sampai 00FF(hex) sudah dipesan oleh sistem operasi (DOS).

**INT** : interrupt. Menginterupsi prosesor.

Prosesor akan:

1. Membackup data registernya saat itu,
2. Menghentikan apa yang sedang dikerjakannya,
3. Melompat ke bagian interrupt-handler (entah dimana kita tidak tahu, sudah ditentukan BIOS dan DOS),
4. Melakukan interupsi,
5. Mengembalikan data registernya,
6. Meneruskan pekerjaan yang tadi ditunda.



sintaks: INT {nomor interupsi}

**IRET** : interrupt-handler return.

Kita bisa membuat interrupt-handler sendiri dengan berbagai cara. perintah IRET adalah perintah yang menandakan bahwa interrupt-handler kita selesai, dan prosesor boleh melanjutkan pekerjaan yang tadi tertunda.

**CALL** : call procedure. Memanggil sebuah prosedur.

sintaks: CALL {label nama prosedur}

**RET** : return. Tanda selesai prosedur.

Setiap prosedur harus memiliki RET di ujungnya.

sintaks: RET ;(saja)

**HLT** : halt. Membuat prosesor menjadi tidak aktif. Prosesor harus mendapat interupsi dari luar atau di-reset supaya aktif kembali. **Jadi, jangan gunakan perintah HLT untuk mengakhiri program!!**

Sintaks: HLT ;(saja). **NOP** : no operation.

Perintah ini memakan 1 byte di memori tetapi tidak menyuruh prosesor melakukan apa-apa selama 3 clock prosesor. Berikut contoh potongan program untuk melakukan *delay* selama 0,1 detik pada prosesor Intel 80386 yang berkecepatan 16 MHz., *mov ECX, 533333334d ;ini adalah bilangan desimal idle: nop loop idle.*