

PRAKTIKUM SISTEM OPERASI
MODUL 3



Nama : NICKY JULYATRIKA SARI

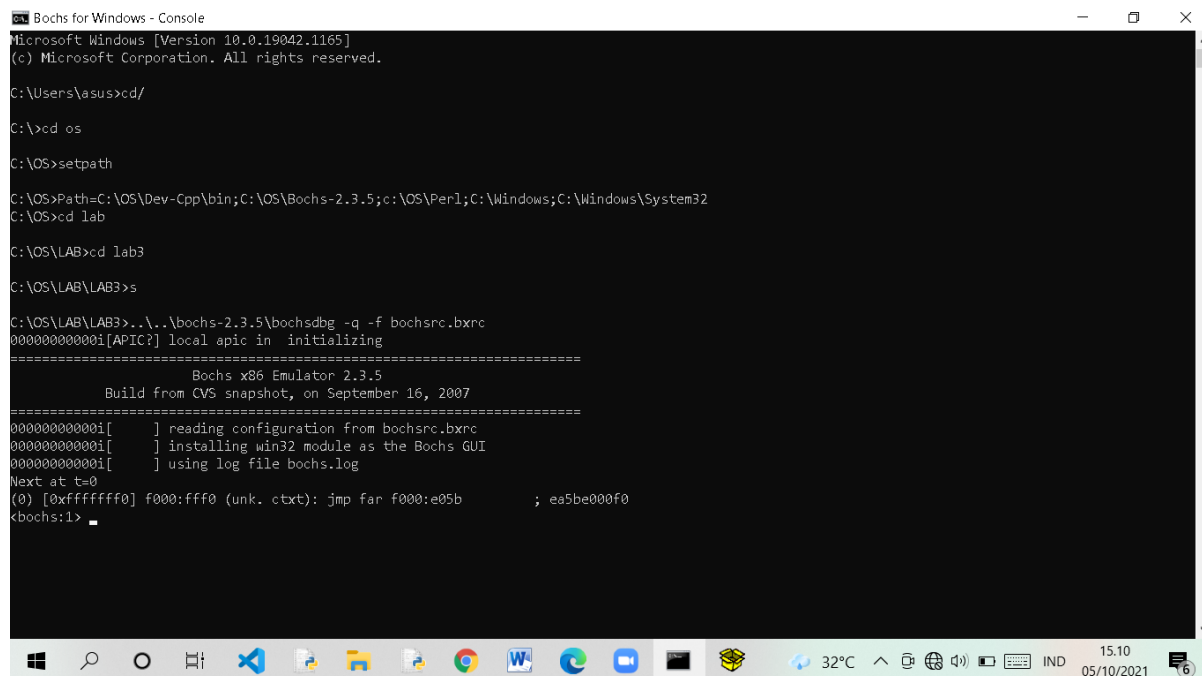
NIM : L200200101

PROGRAM STUDI
INFORMATIKA

FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
TAHUN 2021/2022

1. Langkah kerja

1. 'Start|run' ketik 'cmd' lanjutkan dengan 'CD OS', 'setpath' dan 'cd LAB/LAB3'.
2. File sudah di siapkan, tugas anda adalah melakukan proses 'debugging'. Program 'Bochs' yang diaktifkan adalah program versi debug yaitu 'Bochsdbg', lihatlah pada perintah yang tersimpan pada file 's.bat', ketika 'type s.bat'.
3. Mulai melakukan 'debugging': masukan perintah 'S' . Layar pada PC-Simulator akan terlihat gelap, tidak ada aktifitas, tidak ada kesalahan disana tetapi jalannya program dihentikan oleh 'Bochs' menunggu masukan dari user. Pindahlah ke window 'Command Prompt' yang sekarang muncul tampilan seperti pada gambar berikut:



```
Bochs for Windows - Console
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\asus>cd/

C:\>cd os

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

C:\OS\LAB>cd lab3

C:\OS\LAB\LAB3>s

C:\OS\LAB\LAB3>..\bochs-2.3.5\bochsdbg -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 bochs.log
Next at t=0
(0) [0xffffffff] f000:ffff (unk. ctxt): jmp far f000:e05b      ; ea5be000f0
<bochs:1>
```

4. melihat isi register CS dan IP dengan perintah 'r'. Sekarang ketikan 'r' akan ditampilkan teks berikut :

```
Bochs for Windows - Console
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\asus>cd/

C:\>cd os

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

C:\OS\LAB>cd lab3

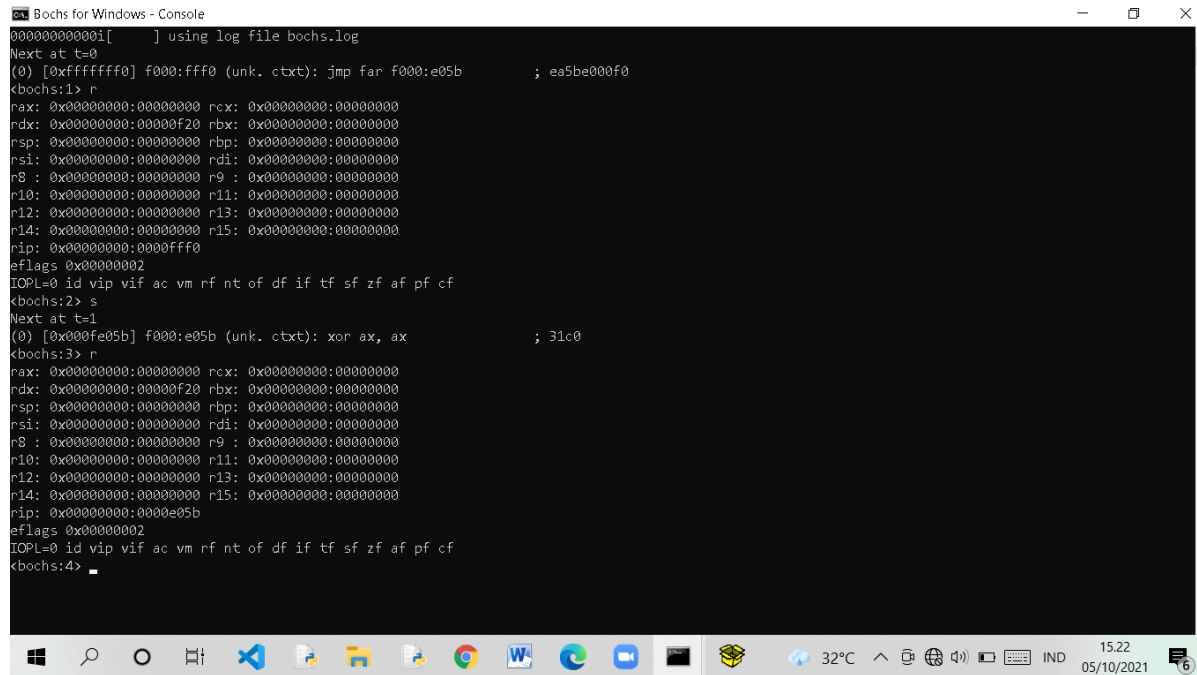
C:\OS\LAB\LAB3>s

C:\OS\LAB\LAB3>..\..\bochs-2.3.5\bochsrc -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 bochs.log
Next at t=0
(0) [0xffffffff] f000:fff0 (unk. ctxt): jmp far f000:e05b ; ea5be00f0
<bochs:1> r
rax: 0x00000000:00000000 rcx: 0x00000000:00000000
rdx: 0x00000000:00000f20 rbx: 0x00000000:00000000
rsp: 0x00000000:00000000 rbp: 0x00000000:00000000
rsi: 0x00000000:00000000 rdi: 0x00000000:00000000
r8 : 0x00000000:00000000 r9 : 0x00000000:00000000
r10: 0x00000000:00000000 r11: 0x00000000:00000000
```

5. Selanjutnya kita suruh PC untuk mengeksekusi perintah tersebut, ketikan ‘s’ kemudian lanjutkan dengan perintah ‘r’ . Pada layar akan ditampilkan terks berikut:

```
Bochs for Windows - Console
00000000000i[ ] using log file bochs.log
Next at t=0
(0) [0xffffffff] f000:fff0 (unk. ctxt): jmp far f000:e05b ; ea5be00f0
<bochs:1> r
rax: 0x00000000:00000000 rcx: 0x00000000:00000000
rdx: 0x00000000:00000f20 rbx: 0x00000000:00000000
rsp: 0x00000000:00000000 rbp: 0x00000000:00000000
rsi: 0x00000000:00000000 rdi: 0x00000000:00000000
r8 : 0x00000000:00000000 r9 : 0x00000000:00000000
r10: 0x00000000:00000000 r11: 0x00000000:00000000
r12: 0x00000000:00000000 r13: 0x00000000:00000000
r14: 0x00000000:00000000 r15: 0x00000000:00000000
rip: 0x00000000:0000fff0
eflags 0x00000002
IOPL=0 id vip vif ac vm rf nt of df if tf sf zf af pf cf
<bochs:2>
```

6. Selanjutnya kita suruh PC untuk mengeksekusi perintah tersebut, ketikan 's' kemudian lanjutkan dengan perintah 'r'. Pada layar akan ditampilkan terks berikut



```
Bochs for Windows - Console
00000000000i[ ] using log file bochs.log
Next at t=0
(0) [0xffffffff] f000:fff0 (unk. ctxt): jmp far f000:e05b ; ea5be00f0
<bochs:1> r
rax: 0x00000000:00000000 rcx: 0x00000000:00000000
rdx: 0x00000000:00000f20 rbx: 0x00000000:00000000
rsp: 0x00000000:00000000 rbp: 0x00000000:00000000
rsi: 0x00000000:00000000 rdi: 0x00000000:00000000
r8 : 0x00000000:00000000 r9 : 0x00000000:00000000
r10: 0x00000000:00000000 r11: 0x00000000:00000000
r12: 0x00000000:00000000 r13: 0x00000000:00000000
r14: 0x00000000:00000000 r15: 0x00000000:00000000
rip: 0x00000000:0000fff0
eflags 0x00000002
IOPL=0 id vip vif ac vm rf nt of df if tf sf zf af pf cf
<bochs:2> s
Next at t=1
(0) [0x000fe05b] f000:e05b (unk. ctxt): xor ax, ax ; 31c0
<bochs:3> r
rax: 0x00000000:00000000 rcx: 0x00000000:00000000
rdx: 0x00000000:00000f20 rbx: 0x00000000:00000000
rsp: 0x00000000:00000000 rbp: 0x00000000:00000000
rsi: 0x00000000:00000000 rdi: 0x00000000:00000000
r8 : 0x00000000:00000000 r9 : 0x00000000:00000000
r10: 0x00000000:00000000 r11: 0x00000000:00000000
r12: 0x00000000:00000000 r13: 0x00000000:00000000
r14: 0x00000000:00000000 r15: 0x00000000:00000000
rip: 0x00000000:0000e05b
eflags 0x00000002
IOPL=0 id vip vif ac vm rf nt of df if tf sf zf af pf cf
<bochs:4> -
```

7. Masukkan perintah berikut 'vb 0:0x7C00' Maksud perintah ini adalah membuat titik pemberhentian (halte) pada alamat 0000:7C00. Selanjutnya kita perintahkan PC untuk melanjutkan pekerjaannya sekarang, yaitu melanjutkan program yang terdapat pada BIOS untuk memeriksa RAM dan peralatan lainnya. Masukkan perintah 'c'

```
Bochs for Windows - Console
r8 : 0x00000000:00000000 r9 : 0x00000000:00000000
r10: 0x00000000:00000000 r11: 0x00000000:00000000
r12: 0x00000000:00000000 r13: 0x00000000:00000000
r14: 0x00000000:00000000 r15: 0x00000000:00000000
rip: 0x00000000:0000ffff
eflags 0x00000002
IOPL=0 id vip vif ac vm rf nt of df if tf sf zf af pf cf
<bochs:2> s
Next at t=1
(0) [0x000fe05b] f000:e05b (unk. ctxt): xor ax, ax ; 31c0
<bochs:3> r
rax: 0x00000000:00000000 rcx: 0x00000000:00000000
rdx: 0x00000000:00000f20 rbx: 0x00000000:00000000
rsp: 0x00000000:00000000 rbp: 0x00000000:00000000
rsi: 0x00000000:00000000 rdi: 0x00000000:00000000
r8 : 0x00000000:00000000 r9 : 0x00000000:00000000
r10: 0x00000000:00000000 r11: 0x00000000:00000000
r12: 0x00000000:00000000 r13: 0x00000000:00000000
r14: 0x00000000:00000000 r15: 0x00000000:00000000
rip: 0x00000000:0000e05b
eflags 0x00000002
IOPL=0 id vip vif ac vm rf nt of df if tf sf zf af pf cf
<bochs:4> vb 0:0x7C00
<bochs:5> c
(10264512) Breakpoint 10285608, in 0000:7c00 (0x00007c00)
Next at t=2082128
(0) [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00
<bochs:6>
```

- a. pada PC Simulator mulai ditampilkan teks yang berakhir dengan tulisan berikut

```
e-Book_Modul Sistem Operasi (1 x)
Bochs for Windows - Display
Plex86/Bochs UGABios 0.6a 19 Aug 2006
This UGA/VBE Bios is released under the GNU LGPL

Please visit :
. http://bochs.sourceforge.net
. http://www.nongnu.org/vgabios

Bochs VBE Display Adapter enabled

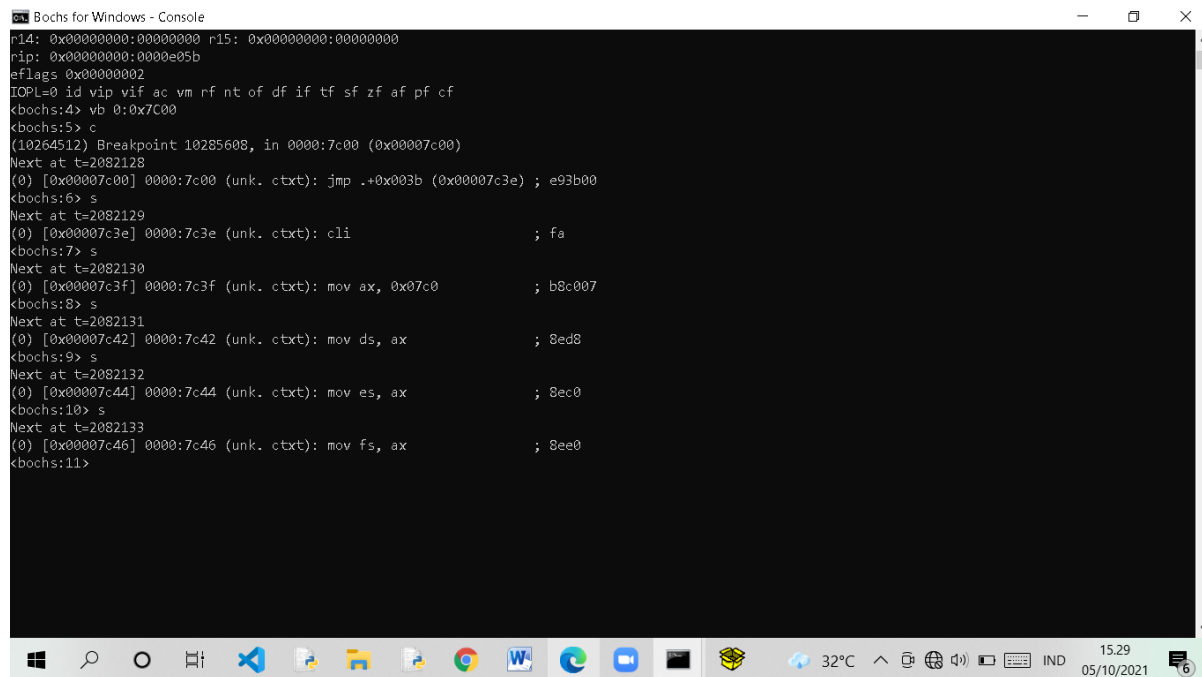
Bochs BIOS - build: 09/10/07
$Revision: 1.183 $ $Date: 2007/09/10 20:00:29 $
Options: apmbios pcbios eltorito rombios32

Booting from Floppy...

program yang berasal dari BOOTSECTOR pada diskboot.
```

8. Selanjutnya bandingkan 10 instruksi berikutnya yang akan dieksekusi oleh PC dengan program yang terdapat pada 'boot.asm', lakukan dengan cara menjalankan PC langkah demi langkah (debugging) dengan perintah 's', tulislah setiap teks yang ditampilkan pada setiap

langkah. Perhatikan semua perubahan yang terjadi. Jika terjadi kesalahan, anda dapat menghentikan proses dengan perintah ‘q’ dan mengulangi proses dari awal atau klik tombol POWER pada PC Simulator. Untuk melihat daftar menu yang disediakan dapat digunakan perintah ‘h’. Untuk memerintahkan PC agar melanjutkan prosesnya masukan ‘c’



```
Bochs for Windows - Console
r14: 0x00000000:00000000 r15: 0x00000000:00000000
rip: 0x00000000:0000e05b
eflags 0x00000002
IOPL=0 id vip vif ac vm rf nt of df if tf sf zf af pf cf
<bochs:4> vb 0:0x7C00
<bochs:5> c
(10264512) Breakpoint 10285608, in 0000:7c00 (0x00007c00)
Next at t=20821128
(0) [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00
<bochs:6> s
Next at t=20821129
(0) [0x00007c3e] 0000:7c3e (unk. ctxt): cli ; fa
<bochs:7> s
Next at t=20821130
(0) [0x00007c3f] 0000:7c3f (unk. ctxt): mov ax, 0x07c0 ; b8c007
<bochs:8> s
Next at t=20821131
(0) [0x00007c42] 0000:7c42 (unk. ctxt): mov ds, ax ; 8ed8
<bochs:9> s
Next at t=20821132
(0) [0x00007c44] 0000:7c44 (unk. ctxt): mov es, ax ; 8ec0
<bochs:10> s
Next at t=20821133
(0) [0x00007c46] 0000:7c46 (unk. ctxt): mov fs, ax ; 8ee0
<bochs:11>
```

9. Menghentikan PC Simulator pada saat akan menjalankan program ‘kernel.bin’: Mulailah dari awal, hentikan ‘debugging’ sebelumnya dengan memasukkan perintah ‘q’, jika PC Simulator sedang bekerja, hentikan dengan menekan tombol ‘CTRL+C’ kemudian ‘q’. Selanjutnya mulai dari awal, ketik ‘s’ (dari ‘Command Prompt’).

```
Bochs for Windows - Console
Bochs is exiting. Press ENTER when you're ready to close this window.
Terminate batch job (Y/N)? y
C:\OS\LAB\LAB3>s
C:\OS\LAB\LAB3>..\..\bochs-2.3.5\bochsrc -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 bochs.log
Next at t=0
(0) [0xffffffff] f000:fff0 (unk. ctxt): jmp far f000:e05b ; ea5be00f0
<bochs:1> _
```

10. Kemudian buatlah break-point, masukan perintah 'vb 0x0100:0x0000' untuk menghentikan langkah saat PC mulai mengeksekusi instruksi dari program 'kernel.bin'.

```
Bochs for Windows - Console
00000000000i[ ] using log file bochs.log
Next at t=0
(0) [0xffffffff] f000:fff0 (unk. ctxt): jmp far f000:e05b ; ea5be00f0
<bochs:1> vb 0x0100:0x0000
<bochs:2>
```

11. Selanjutnya perintahkan PC untuk melanjutkan pekerjaan, 'c'

```
Bochs for Windows - Console
000000000000i[ ] using log file bochs.log
Next at t=0
(0) [0xffffffff] f000:ffff (unk. ctxt): jmp far f000:e05b ; ea5be00f0
<bochs:1> vb 0x0100:0x0000
<bochs:2> c
(10264512) Breakpoint 10285608, in 0100:0000 (0x00001000)
Next at t=2945013
(0) [0x00001000] 0100:0000 (unk. ctxt): mov ax, 0x0100 ; b80001
<bochs:3>
```

12.Selanjutnya teruskan langkah PC Simulator step-by-step minimal sebanyak 10x, ketik ‘s’, step berikutnya dapat dilakukan dengan cara menekan tombol secara langsung. Perhatikan dan catat setiap perubahan teks yang ditampilkan. Bandingkan dengan source-code pada program ‘kernel.asm’

```
Bochs for Windows - Console
(10264512) Breakpoint 10285608, in 0100:0000 (0x00001000)
Next at t=2945013
(0) [0x00001000] 0100:0000 (unk. ctxt): mov ax, 0x0100 ; b80001
<bochs:3> s
Next at t=2945014
(0) [0x00001003] 0100:0003 (unk. ctxt): mov ds, ax ; 8ed8
<bochs:4> s
Next at t=2945015
(0) [0x00001005] 0100:0005 (unk. ctxt): mov es, ax ; 8ec0
<bochs:5> s
Next at t=2945016
(0) [0x00001007] 0100:0007 (unk. ctxt): cli ; fa
<bochs:6> s
Next at t=2945017
(0) [0x00001008] 0100:0008 (unk. ctxt): mov ss, ax ; 8ed0
<bochs:7> s
Next at t=2945018
(0) [0x0000100a] 0100:000a (unk. ctxt): mov sp, 0xffff ; bcffff
<bochs:8> s
Next at t=2945019
(0) [0x0000100d] 0100:000d (unk. ctxt): sti ; fb
<bochs:9> s
Next at t=2945020
(0) [0x0000100e] 0100:000e (unk. ctxt): push dx ; 52
<bochs:10> s
Next at t=2945021
(0) [0x0000100f] 0100:000f (unk. ctxt): push es ; 06
<bochs:11> s
Next at t=2945022
(0) [0x00001010] 0100:0010 (unk. ctxt): xor ax, ax ; 31c0
<bochs:12> s
Next at t=2945023
(0) [0x00001012] 0100:0012 (unk. ctxt): mov es, ax ; 8ec0
<bochs:13>
```


Tugas

1. Buatlah tabel pemetaan memori pada PC selengkap mungkin.

- tabel pemetaan memori

a. Menurut Blok Memory

Blok Memory	Alokasi Pemakaian
F 0 0 0 0	ROM BIOS,Diagnostic,Basic
E 0 0 0 0	ROM program
D 0 0 0 0	ROM program
C 0 0 0 0	Perluasan BIOS untuk hardisk XT
B 0 0 0 0	Monokrom Monitor
A 0 0 0 0	Monitor EGA,VGS,dll
9 0 0 0 0	Daerah kerja pemakai s/d 640kb
8 0 0 0 0	Daerah kerja pemakai s/d 576kb
7 0 0 0 0	Daerah kerja pemakai s/d 512kb
6 0 0 0 0	Daerah kerja pemakai s/d 448kb
5 0 0 0 0	Daerah kerja pemakai s/d 384kb
4 0 0 0 0	Daerah kerja pemakai s/d 320kb
3 0 0 0 0	Daerah kerja pemakai s/d 256kb
2 0 0 0 0	Daerah kerja pemakai s/d 192kb
1 0 0 0 0	Daerah kerja pemakai s/d 128kb
0 0 0 0 0	Daerah kerja pemakai s/d 64kb

b. Menurut Alamat Fisik

Alamat Fisik(heks)	Keterangan
0 0 0 0 - 0 0 0 7 F	BIOS Interrupt Vektor
0 0 0 0 - 0 0 3 F F	DOS Interrupt Vektor
0 0 4 0 0 - 0 0 4 F F	BIOS Data Area
0 0 5 0 0 - 0 0 5 F F	DOS dan Basic Data Area
0 0 6 0 0 - 9 C 6 F F	RAM Working Space
A 0 0 0 - A 3 F F F	Cadangan
A 4 0 0 0 - A F F F F	Monochrome Adapter
B 0 0 0 0 - B 1 0 0 0	Video Buffer
B 1 0 0 1 - B 7 F F F	Color/Grafics Adapter
B 8 0 0 0 - B B F F F	
B C 0 0 0 - B F F F F	
C 0 0 0 0 - C 7 F F F	Eksapansi Memory
C 8 0 0 0 - F 3 F F F	Hardisk
C C 0 0 0 - F 3 F F F	Cadangan
F 4 0 0 0 - F 5 F F F	User ROM(8k)

F 6 0 0 0 - F D F F F	ROM Basic(32k)
F E 0 0 0 - F F F F F	ROM BIOS(8k)

2. Baca buku referensi, jelaskan perbedaan antara mode kerja 'Real-Mode' dan mode kerja 'Protect-Mode' pada PC IBM Compatible.

- REAL-MODE

Real Mode adalah suatu modus dimana prosesor intel x86 berlangsung seolah olah dirinya adalah suatu prosesor 8085 atau intel 8088,meski ia merupakan prosesor intel 80286 atau lebih tinggi. Maka dari itu modus ini juga disebut sebagai modus 8086 (8086 Mode).Didalam modus ini ,prosesor hanyalah mampu mengeksekusi intruksi 16-Bit saja bersama dengan mengenakan register internal yang berukuran 16-Bit ,dan juga sekedar bisa terhubung Cuma 1024KB berasal dari memori gara gara sekedar kenakan 20-Bit jalur bus alamat.Seluruh program DOS terjadi terhadap modus ini.Prosesor yang dirilis sehabis 8085 ,semacam Intel 80286 juga mampu menjalankan intruksi 16-Bit ,tetapi jauh lebih cepat dibandingkan 8085.bersama dengan kata lain ,Intel 80286 sah-sah benar kompetibel bersama prosesor Intel 8086 yang didesain sebelumnya .Agar prosesor Intel 80286 pun sanggup menjalankan program-program 16-Bit yang didesain untuk 8085(Ibm Pc),bersama tentunya kecepatan yang jauh lebih tinggi.

Didalam Real-Mode ,bukan tersedia proteksi ruang alamat memori,supaya bukan sannggup lakukan multi-tasking.Inilah sebabnya,mengapa program-program DOS bersifat single-tasking ,jikalau didalam modus real terdapat multi-tasking,maka bisa saja besar antara dua program yang tengah terjadi ,berjalan tabrakan(Crash) antara satu bersama lainnya.

- PROTECT-MODE

Modus terproteksi(Protectec Mode) adalah suatu modus dimana terdapat proteksi ruang alamat memori yang ditawarkan oleh mikroskop untuk digunakan oleh platform operasi.Modus

ini berkunjung bersama mikroskop Intel 80286 atau yang lebih tinggi. Sebab punya proteksi ruang alamat memori. maka didalam modus ini platform operasi bisa dilaksanakan multitasking. prosesor Intel 80286 memang dilengkapi kemampuan masuk ke dalam modus terproteksi, namun bukan mampu keluar berasal dari modus itu tanpa wajib mengalami reset (Warm boot atau cold boot). Kesalahan ini udah diperbaiki oleh Intel bersama merilis prosesor Intel 80386 yang mampu masuk kedalam modus terproteksi dan keluar darinya tanpa perlu jalankan reset. Inilah sebabnya mengapa Windows 95/98 dilengkapi persama dengan modus Restart in Ms-Dos Mode, meski sebenarnya platform operasi itu merupakan platform operasi yang berlangsung didalam modus terproteksi.