

**LAPORAN PRAKTIKUM  
SISTEM OPERASI  
MODUL 3**



**DISUSUN OLEH:**

<b>NIM</b>	<b>L200220277</b>
<b>NAMA</b>	<b>MHD. FARHAN LUBIS</b>
<b>KELAS</b>	<b>F</b>

**PROGRAM STUDI INFORMATIKA  
FAKULTAS KOMUNIKASI DAN INFORMATIKA  
UNIVERSITAS MUHAMMADIYAH SURAKARTA**

**2023**

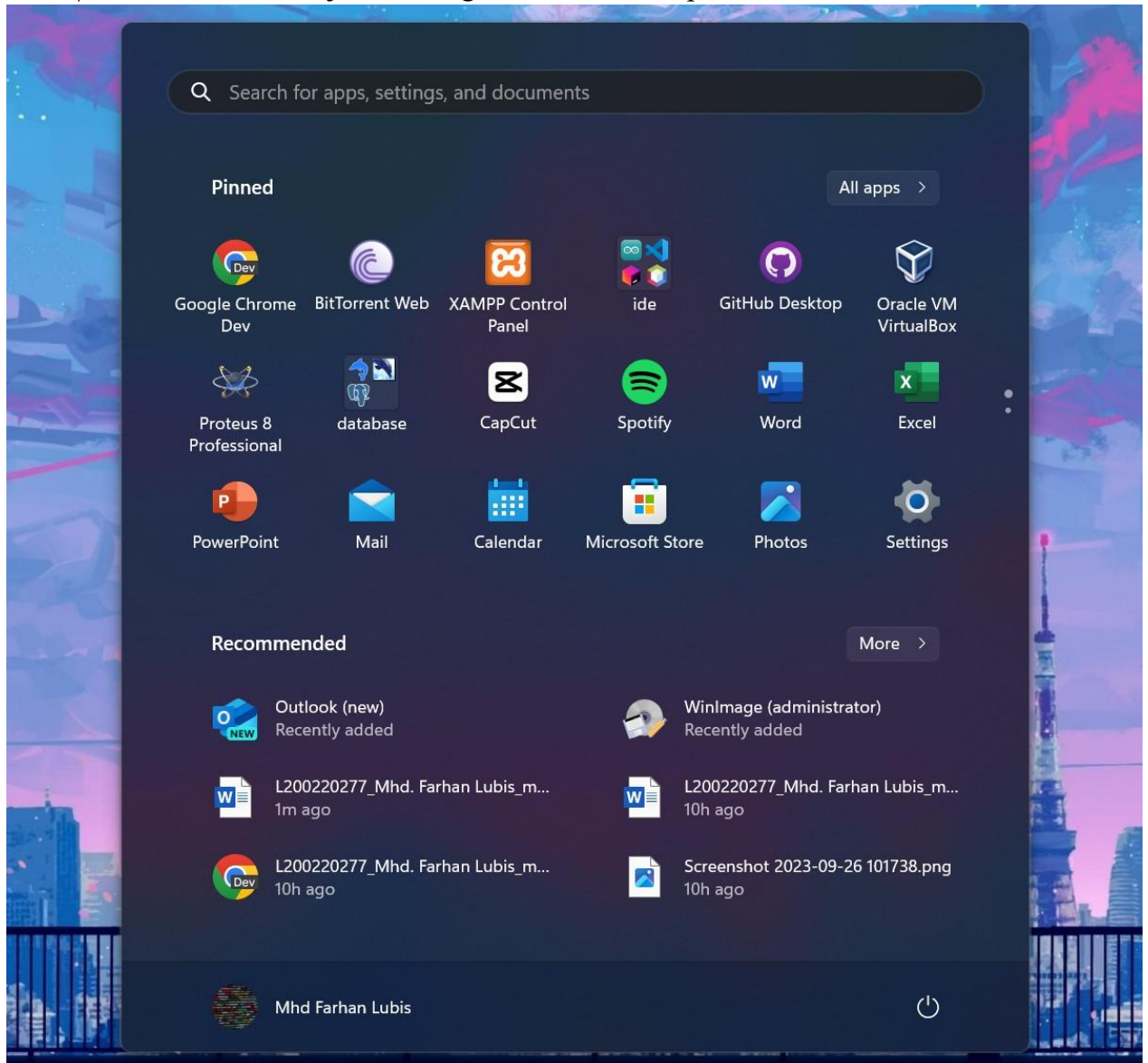
# DAFTAR ISI

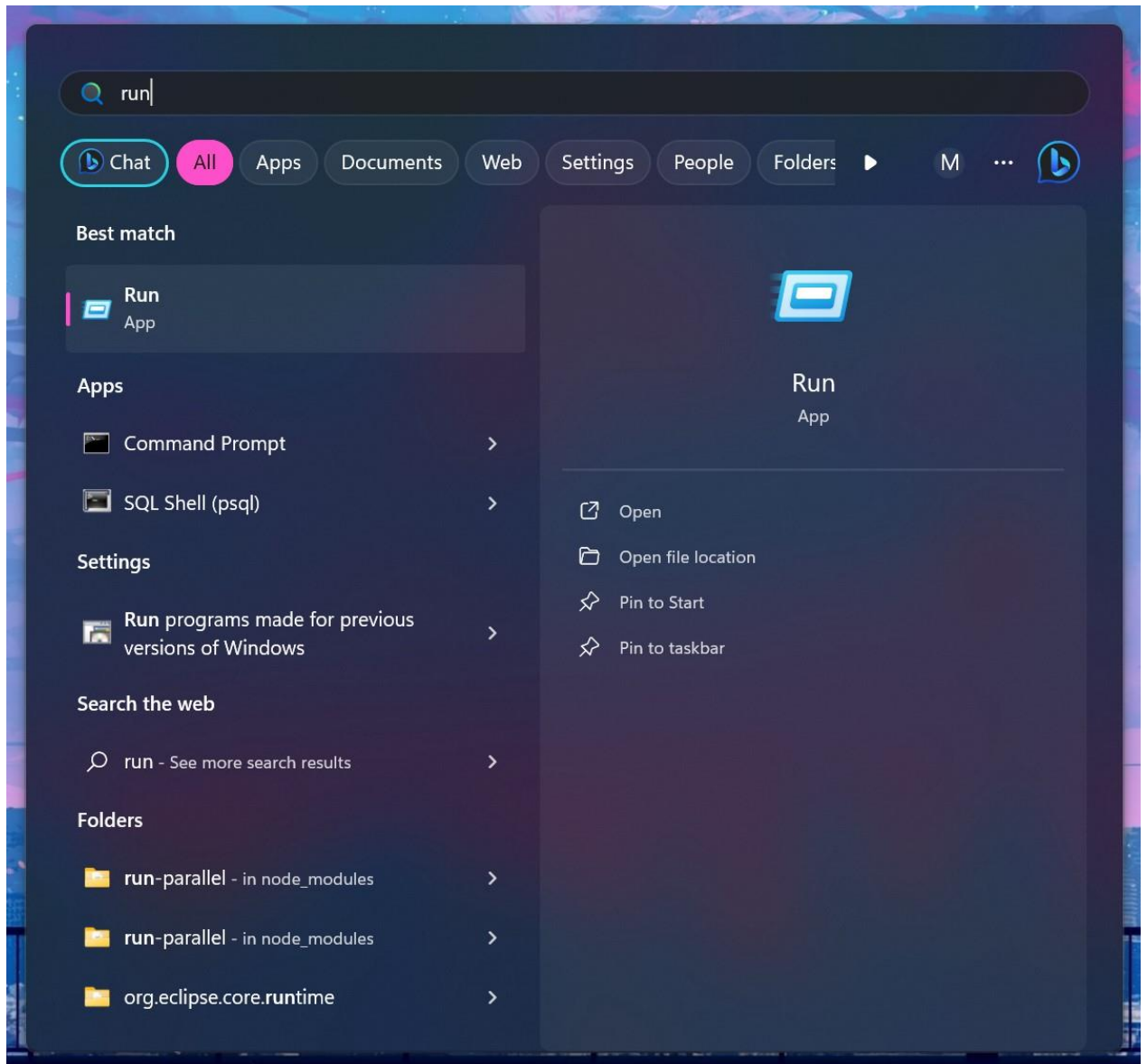
<b>DAFTAR ISI</b> .....	2
<b>LATIHAN</b> .....	4
1. 'Start run' ketik 'cmd' lanjutkan dengan 'CD OS', 'setpath' dan 'cd LAB/LAB3'.....	4
2. Program 'Bochs' yang diaktifkan adalah program versi debug yaitu 'Bochsdbg', Lihat pada perintah yang tersimpan pada file 's.bat', ketika 'type s.bat'.....	7
3. Mulai lakukan 'debugging': dengan memasukan 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. Pindah ke window 'Command Prompt' yang sekarang muncul tampilan seperti pada gambar berikut:.....	10
4. Lihat isi register CS dan IP dengan perintah 'r'. Sekarang ketikan 'r' akan ditampilkan teks berikut : 12	
5. Selanjutnya kita suruh PC untuk mengeksekusi perintah tersebut, ketikan 's' kemudian lanjutkan dengan perintah 'r'. Pada layar akan ditampilkan teks berikut:.....	13
6. Lihat tahapan detail yang dilakukan oleh PC, jalankan perintah 's' secara berulang. Masukan perintah berikut 'vb 0:0x7C00' Maksud perintah ini adalah membuat titik pemberhentian (halte) pada alamat 0000:7C00. Selanjutnya perintahkan PC untuk melanjutkan pekerjaannya sekarang, yaitu melanjutkan program yang terdapat pada BIOS untuk memeriksa RAM dan peralatan lainnya. Masukkan perintah 'c'. Maksud perintah ini adalah teruskan (Continue) prosesnya sampai ke titik pemberhentian. Dalam sekejap PC sudah sampai pada pemberhentian yang dibuat di atas yaitu pada alamat 0000:7C00, pada layar akan tampak teks seperti berikut.....	14
7. Jalankan PC langkah demi langkah (debugging) dengan perintah 's', tuliskan setiap teks yang ditampilkan pada setiap langkah. Instruksi tersebut akan tampak seperti pada gambar berikut:.....	16
8. Tambahkan 'break-point' yang lain (maksimal 7). Tahapan penting berikutnya adalah ketika PC Simulator mulai menjalankan program 'kernel.bin', hal ini terjadi pada alamat '0100:0000'.....	17
9. Hentikan PC Simulator pada saat akan menjalankan program 'kernel.bin': Mulailah dari awal, hentikan 'debugging' sebelumnya dengan memasukan perintah 'q', jika PC Simulator sedang bekerja, hentikan dengan menekan tombol 'CTRL+C' kemudian 'q'. Selanjutnya mulai dari awal, ketik 's'. Kemudian buatlah break-point, masukan perintah 'vb 0x0100:0x0000' untuk menghentikan langkah saat PC mulai mengeksekusi instruksi dari program 'kernel.bin'. Selanjutnya perintahkan PC untuk melanjutkan pekerjaan, 'c'. Jika langkah anda sesuai dengan petunjuk di atas maka pada layar 'debugging' akan di tampilkan teks berikut:.....	18
10. Selanjutnya teruskan langkah PC Simulator step-by-step minimal sebanyak 10x, ketik 's', step berikutnya dapat dilakukan dengan cara menekan tombol secara langsung.....	19
11. Langkah-langkah di atas adalah merupakan salah satu cara yang banyak di gunakan para pengembang perangkat lunak, termasuk pengembangan sistim operasi.....	19
<b>TUGAS</b> .....	19
1. Tabel pemetaan memori pada PC lengkap.....	19

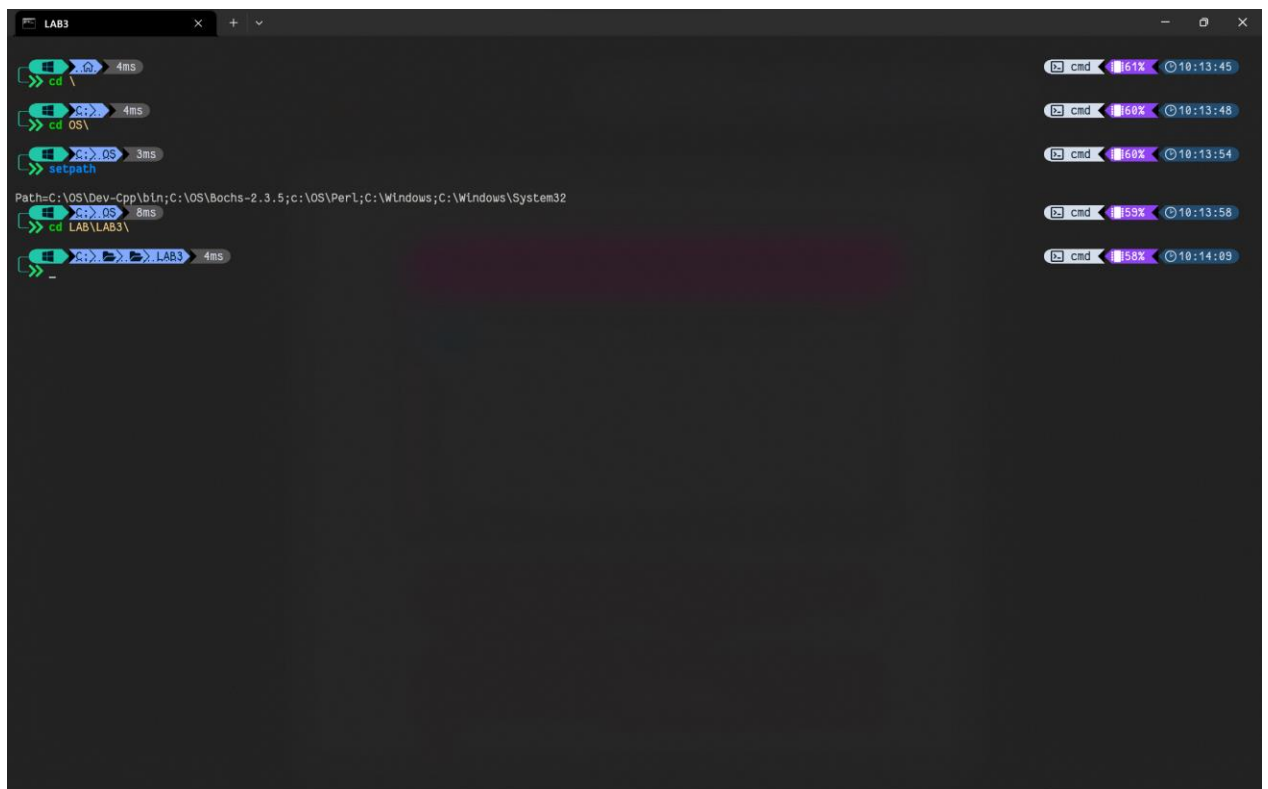
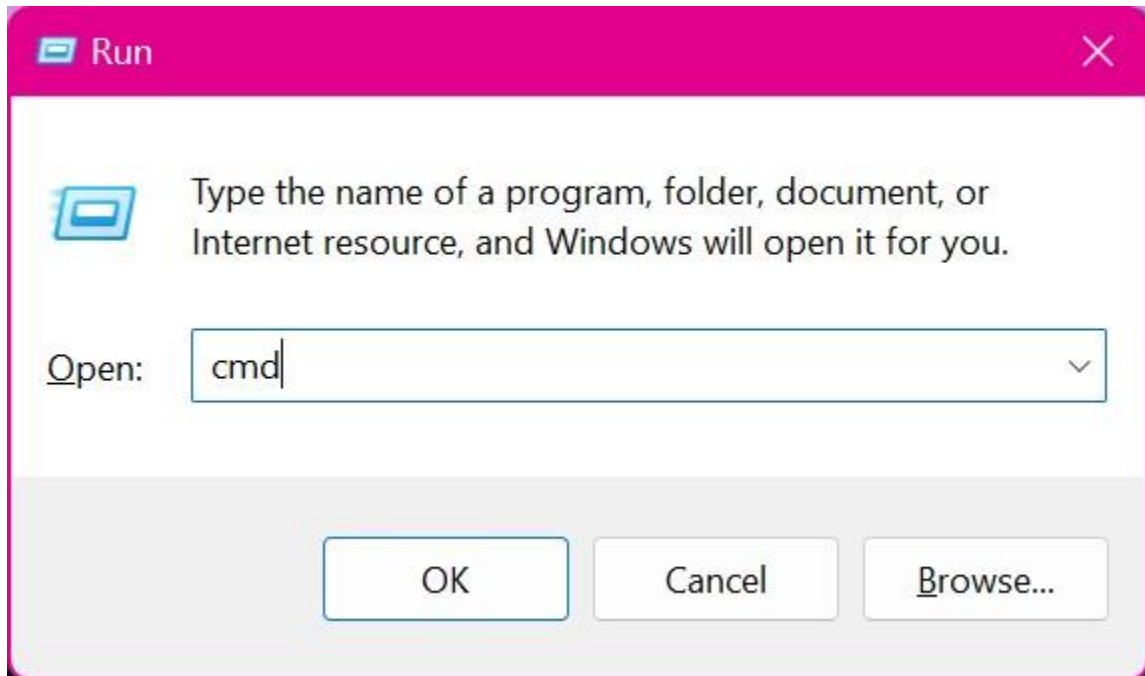
2. Perbedaan antara mode kerja 'Real-Mode' dan mode kerja 'Protect-Mode' pada PC IBM Compatible.....	20
a. Akses Memori .....	20
b. Perlindungan dan Keamanan .....	20
c. Kapasitas Memori .....	20
d. Penggunaan Sumber Daya .....	20
e. Kompatibilitas.....	21

# LATIHAN

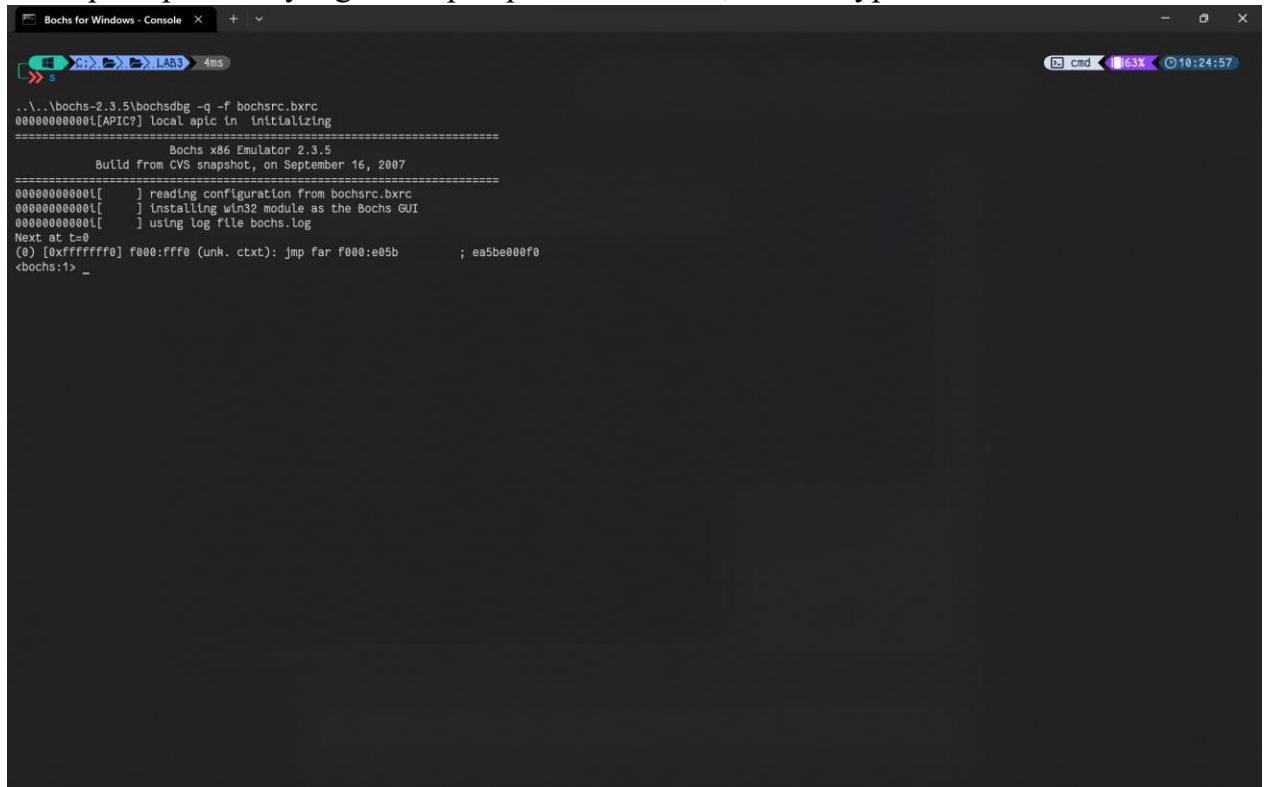
1. 'Start|run' ketik 'cmd' lanjutkan dengan 'CD OS', 'setpath' dan 'cd LAB/LAB3'.



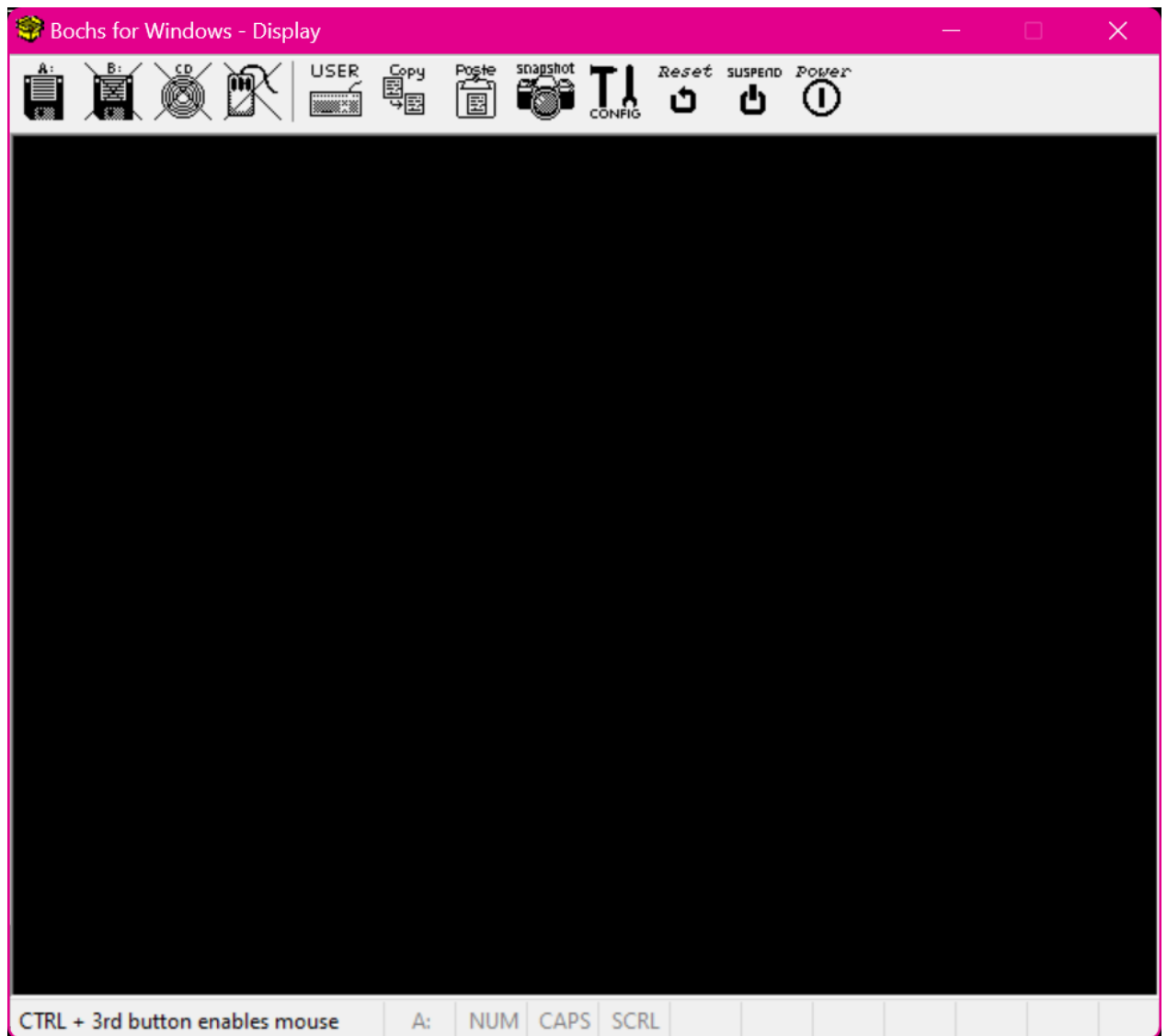




2. Program 'Bochs' yang diaktifkan adalah program versi debug yaitu 'Bochsdbg',  
Lihat pada perintah yang tersimpan pada file 's.bat', ketika 'type s.bat'.



```
Bochs for Windows - Console
C:\Users\LAB3> 4ms
C:\Users\LAB3> s
..\..\bochs-2.3.5\bochsdbg -q -f bochsrc.bxrc
000000000001[APIC?] local apic in initializing
=====
Bochs x86 Emulator 2.3.5
Build from CVS snapshot, on September 16, 2007
=====
000000000001[ ] reading configuration from bochsrc.bxrc
000000000001[ ] installing win32 module as the Bochs GUI
000000000001[ ] using log file bochs.log
Next at t=0
(0) [0xffffffff] f000:ffff (unk. ctxt): jmp far f000:e05b ; ea5be00f0
<bochs:1> _
```



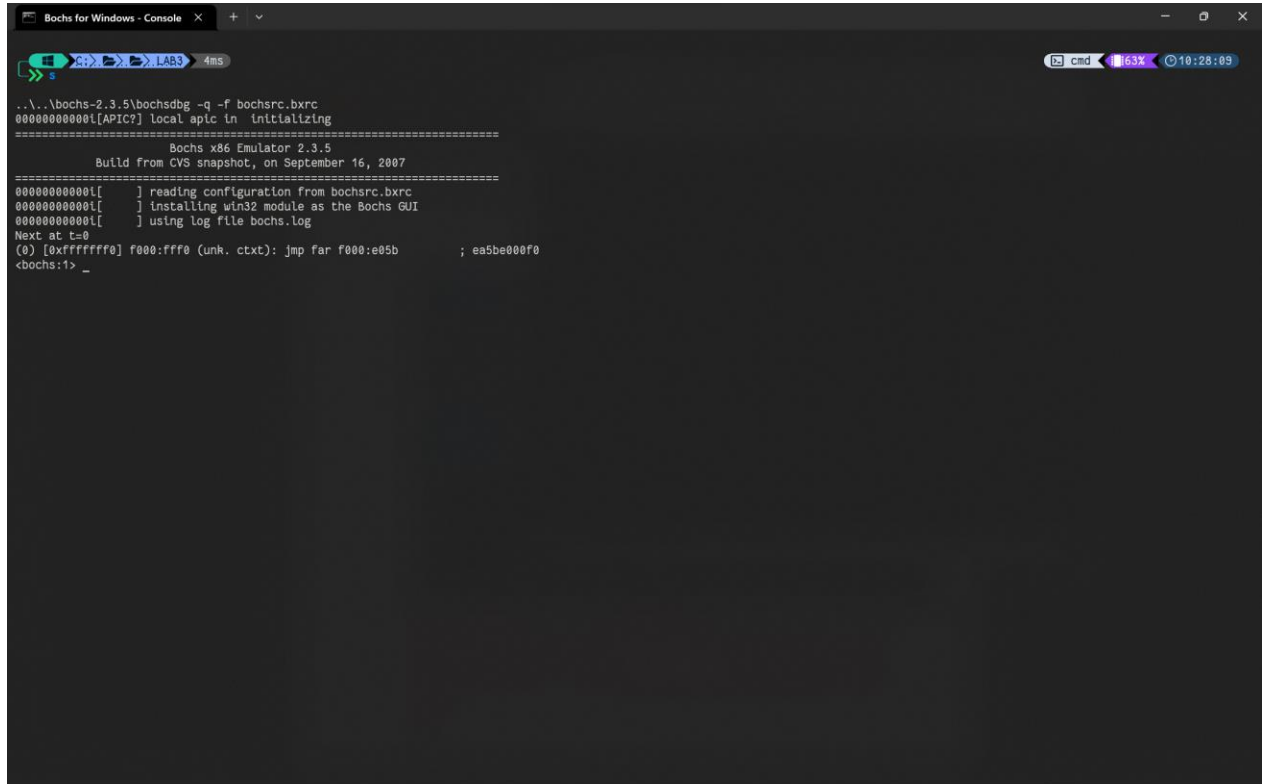


```
LAB3
C:\> .\bochs-2.3.5\bochsdg -q -f bochsrc.bxrc
000000000001[APIC?] local apic in initializing
=====
Bochs x86 Emulator 2.3.5
Build from CVS snapshot, on September 16, 2007
=====
000000000001[ ] reading configuration from bochsrc.bxrc
000000000001[ ] installing win32 module as the Bochs GUI
000000000001[ ] using log file bochs.log
Next at t=0
(0) [0xffffffff] f000:ffff (unk. ctxt): jmp far f000:e05b ; ea5be000f0
<bochs:1> c
=====
Bochs is exiting with the following message:
[VGUI ] POWER button turned off.
=====
# In bx_win32_gui_c::exit(void)!

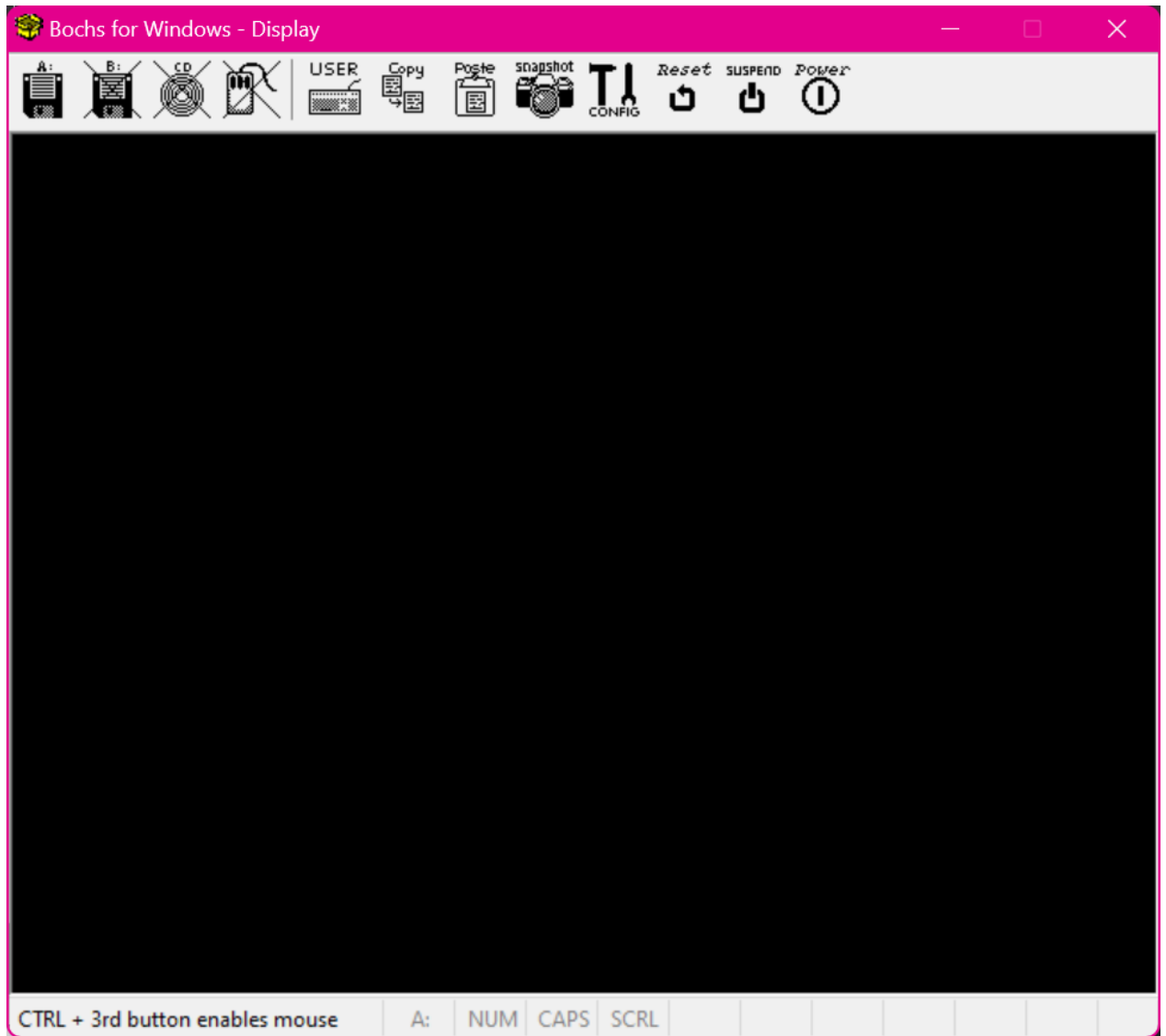
C:\> type s.bat
1m 16s 420ms
..\.bochs-2.3.5\bochsdg -q -f bochsrc.bxrc

C:\>
3ms
```

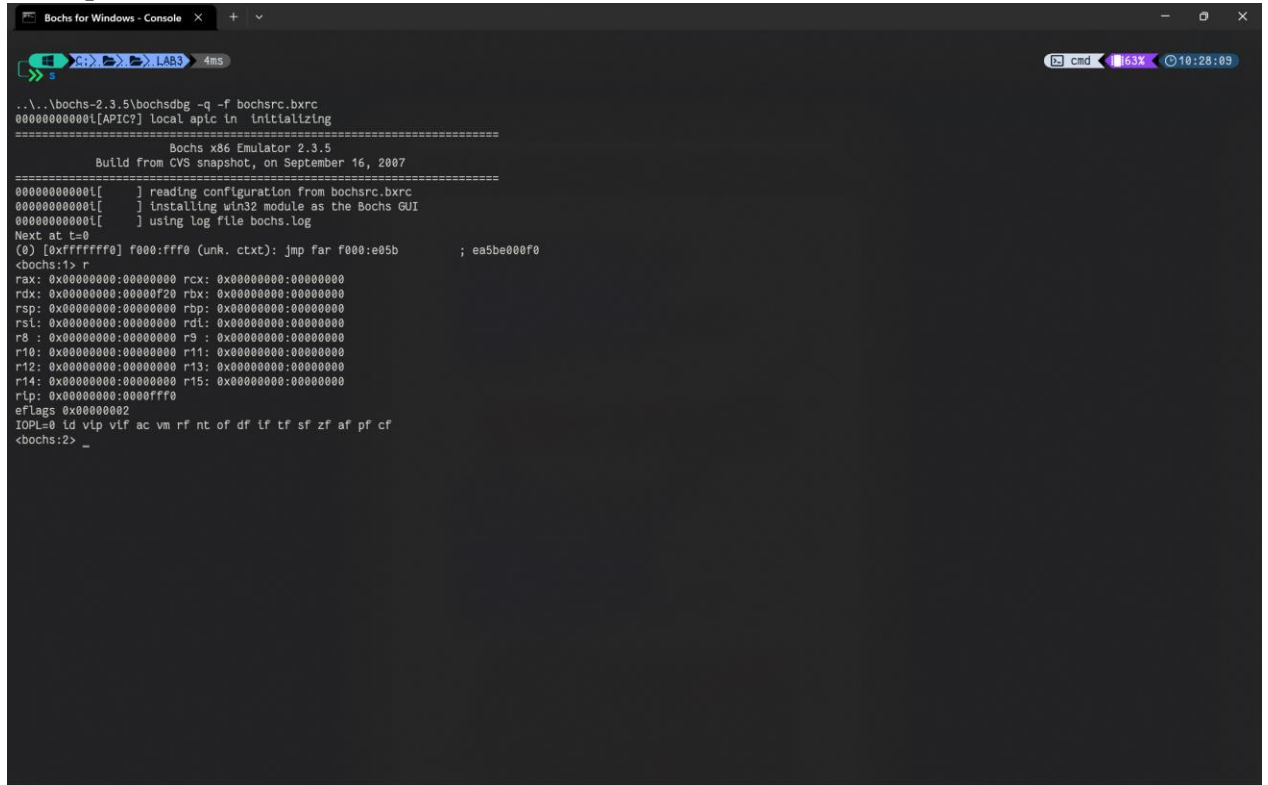
3. Mulai lakukan ‘debugging’: dengan memasukkan perintah ‘S’. Layar pada PC-Simulator akan terlihat gelap, tidak ada aktifitas, tidak ada kesalahan disana tetapi jalannya program dihentikan oleh ‘Bochs’ menunggu masukan dari user. Pindah ke window ‘Command Prompt’ yang sekarang muncul tampilan seperti pada gambar berikut:



```
Bochs for Windows - Console
C:\> cd C:\Users\LAB3\> 4ms
>>> s
...\\bochs-2.3.5\bochsrc -q -f bochsrc.bxrc
000000000001[APIC?] local apic in initializing
=====
Bochs x86 Emulator 2.3.5
Build from CVS snapshot, on September 16, 2007
=====
000000000001[ ] reading configuration from bochsrc.bxrc
000000000001[ ] installing win32 module as the Bochs GUI
000000000001[ ] using log file bochs.log
Next at t=0
(0) [0xffffffff] f000:ffff (unk. ctxt): jmp far f000:e05b ; ea5be00f0
<bochs:1> _
```

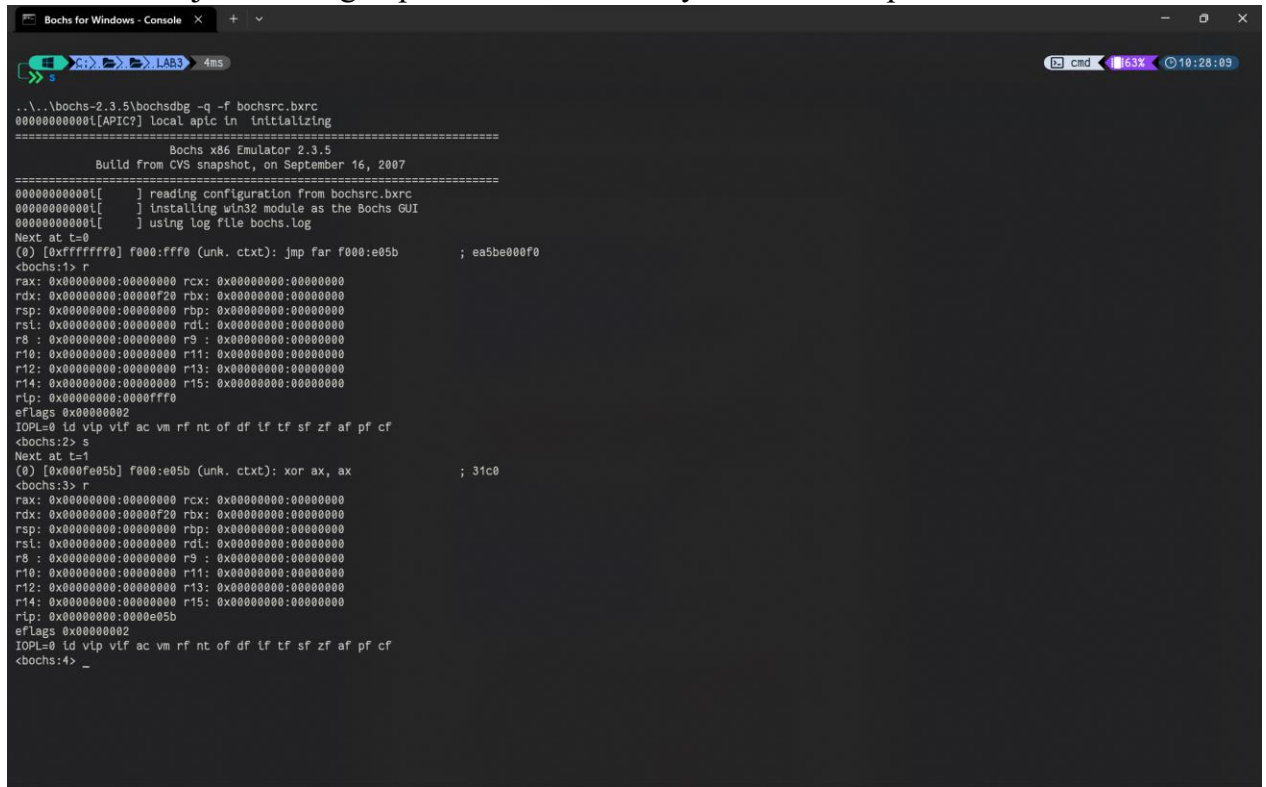


4. Lihat isi register CS dan IP dengan perintah 'r'. Sekarang ketikkan 'r' akan ditampilkan teks berikut :



```
Bochs for Windows - Console
G:\> cd \> cd LAB3 > 4ms
..\\..bochs-2.3.5\bochsrc -q -f bochsrc.bxrc
000000000001[APIC?] local apic in initializing
=====
Bochs x86 Emulator 2.3.5
Build from CVS snapshot, on September 16, 2007
=====
000000000001[ ] reading configuration from bochsrc.bxrc
000000000001[ ] installing win32 module as the Bochs GUI
000000000001[ ] using log file bochs.log
Next at t=0
(0) [0xfffffff0] f000:ffff (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: 0x00000000:0000ffff
eflags 0x00000002
IOPL=0 id vlp vlf ac vm rf nt of df if tf sf zf af pf cf
<bochs:2> _
```

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

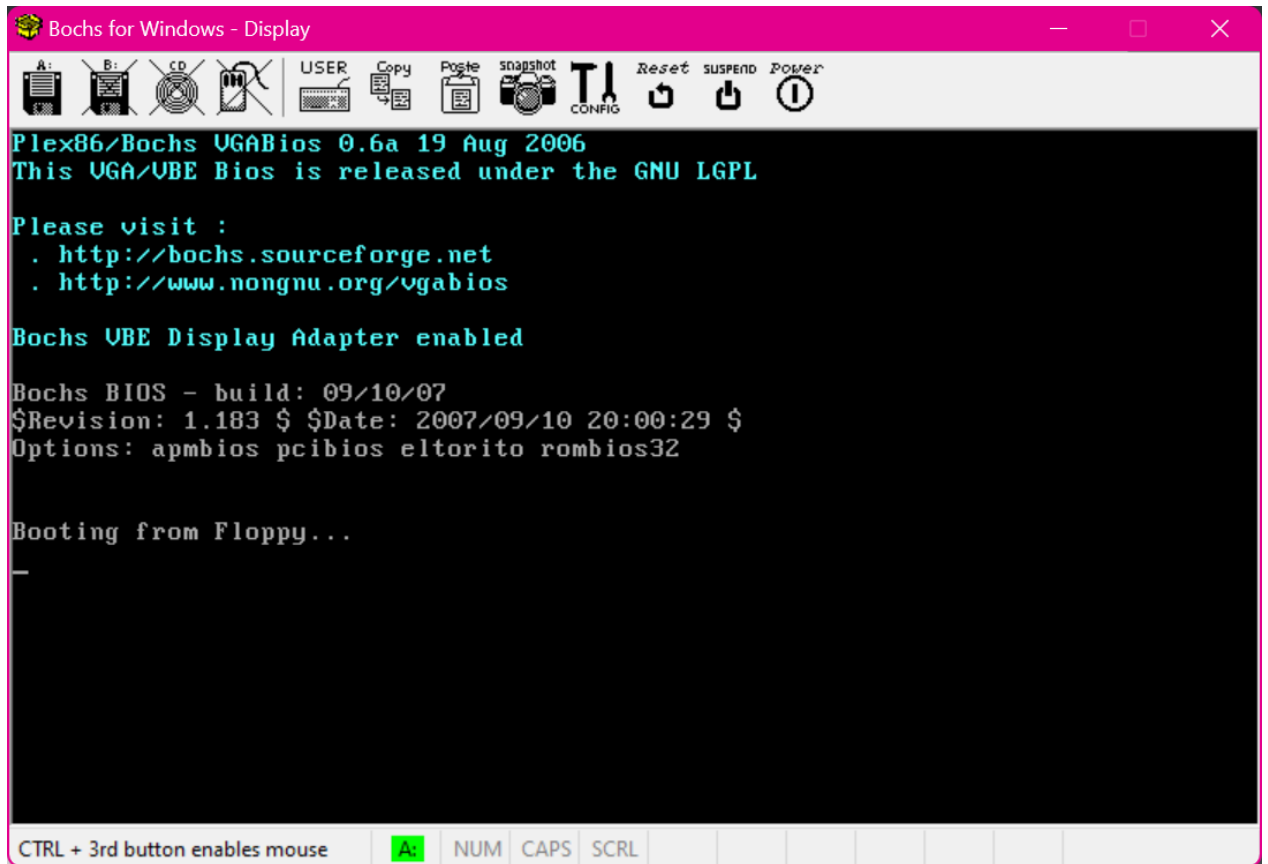


```
Bochs for Windows - Console
C:\Users\LAB3> 4ms
..\\..\\bochs-2.3.5\\bochsrc -q -f bochsrc.bxrc
0000000000000000[APIC?] local apic in initializing
=====
Bochs x86 Emulator 2.3.5
Build from CVS snapshot, on September 16, 2007
=====
0000000000000000[ ] reading configuration from bochsrc.bxrc
0000000000000000[ ] installing win32 module as the Bochs GUI
0000000000000000[ ] using log file bochs.log
Next at t=0
(0) [0xffffffff] f000:ffff (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: 0x00000000:0000ffff
eflags 0x00000002
IOPL=0 id vlp vlf ac vm rf nt of df lf 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 vlp vlf ac vm rf nt of df lf tf sf zf af pf cf
<bochs:4> _
```

6. Lihat tahapan detail yang dilakukan oleh PC, jalankan perintah 's' secara berulang. Masukkan perintah berikut 'vb 0:0x7C00' Maksud perintah ini adalah membuat titik pemberhentian (halte) pada alamat 0000:7C00. Selanjutnya perintahkan PC untuk melanjutkan pekerjaannya sekarang, yaitu melanjutkan program yang terdapat pada BIOS untuk memeriksa RAM dan peralatan lainnya. Masukkan perintah 'c'. Maksud perintah ini adalah teruskan (Continue) prosesnya sampai ke titik pemberhentian. Dalam sekejap PC sudah sampai pada pemberhentian yang dibuat di atas yaitu pada alamat 0000:7C00, pada layar akan tampak teks seperti berikut.

```
Bochs for Windows - Console
000000000001[ ] installing win32 module as the Bochs GUI
000000000001[ ] using log file bochs.log
Next at t=0
(0) [0xfffffff0] f000:ffff (unk. ctxt): jmp far f000:e05b ; ea5be00f0
<bochs:1> r
rax: 0x00000000:00000000 rcx: 0x00000000:00000000
rdx: 0x00000000:0000f20 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: 0x00000000:000ffff0
eflags 0x00000002
IOPL=0 id vip vlf ac vm rf nt of df lf 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: 0x00000000:0000f20 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 vlf ac vm rf nt of df lf tf sf zf af pf cf
<bochs:4> s
Next at t=2
(0) [0x000fe05d] f000:e05d (unk. ctxt): out 0xd, al ; e60d
<bochs:5> s
Next at t=3
(0) [0x000fe05f] f000:e05f (unk. ctxt): out 0xda, al ; e6da
<bochs:6> s
Next at t=4
(0) [0x000fe061] f000:e061 (unk. ctxt): mov al, 0xc0 ; b0c0
<bochs:7> s
Next at t=5
(0) [0x000fe063] f000:e063 (unk. ctxt): out 0xd6, al ; e6d6
<bochs:8> vb 0:0x7C00
<bochs:9> vb 0:0x7C00
<bochs:10> c
(2005679513) Breakpoint 10285608, ln 0000:7c00 (0x00007c00)
Next at t=2086087
(0) [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00
<bochs:11> _
```

Sedangkan pada PC Simulator mulai ditampilkan teks yang berakhir dengan tulisan berikut:



7. Jalankan PC langkah demi langkah (debugging) dengan perintah ‘s’, tuliskan setiap teks yang ditampilkan pada setiap langkah. Instruksi tersebut akan tampak seperti pada gambar berikut:

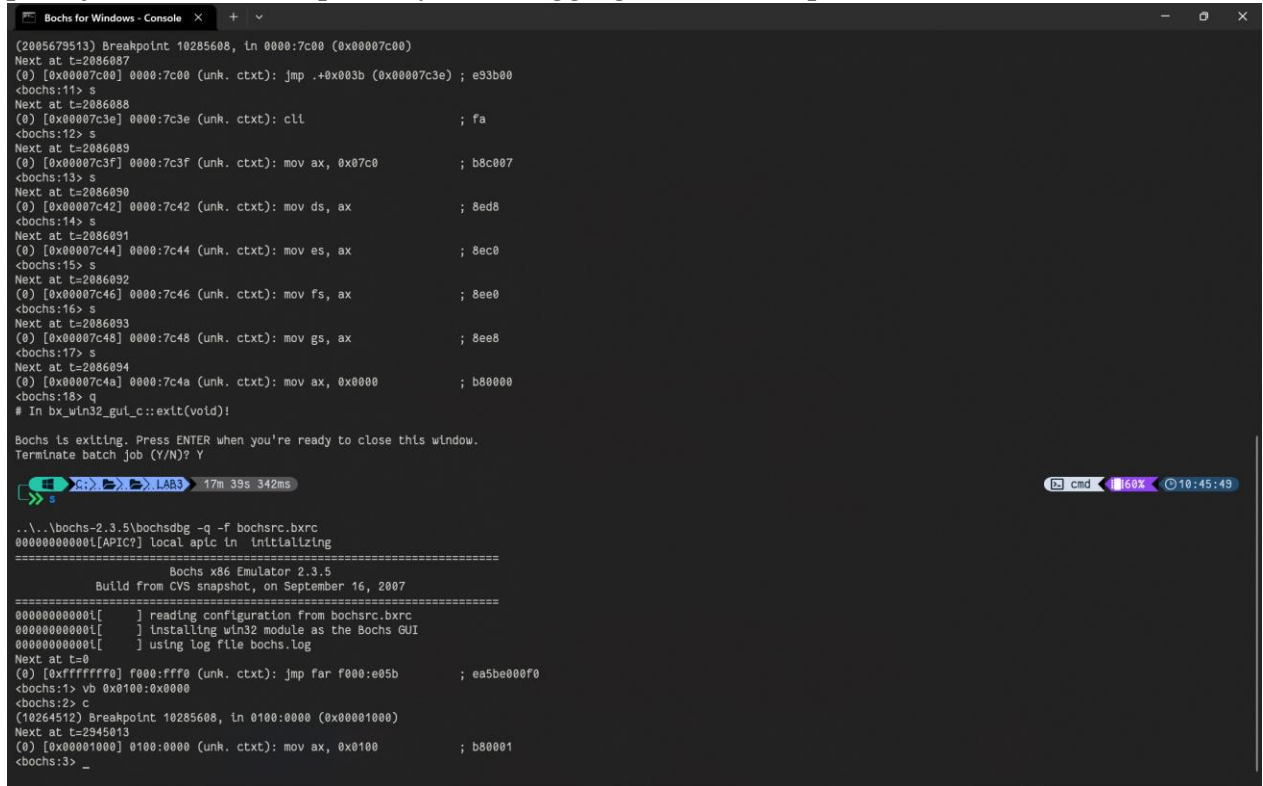
```
Bochs for Windows - Console
IOPL=0 id vlp vlf ac vm rf nt of df lf tf sf zf af pf cf
<bochs:2> s
Next at t=1
(0) [0x000fe05b] f000:e05b (unk. ctxt): xor ax, ax ; 31c0
<bochs:3> r
rax: 0x00000000:00000000 rcx: 0x00000000:00000000
rdx: 0x00000000:00000f20 rbx: 0x00000000:00000000
rsp: 0x00000000:00000000 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 vlp vlf ac vm rf nt of df lf tf sf zf af pf cf
<bochs:4> s
Next at t=2
(0) [0x000fe05d] f000:e05d (unk. ctxt): out 0xd, al ; e60d
<bochs:5> s
Next at t=3
(0) [0x000fe05f] f000:e05f (unk. ctxt): out 0xda, al ; e6da
<bochs:6> s
Next at t=4
(0) [0x000fe061] f000:e061 (unk. ctxt): mov al, 0xc0 ; b0c0
<bochs:7> s
Next at t=5
(0) [0x000fe063] f000:e063 (unk. ctxt): out 0xd6, al ; e6d6
<bochs:8> vb 0:0x7c00
<bochs:9> vb 0:0x7c00
<bochs:10> c
(2005679513) Breakpoint 10285608, ln 0000:7c00 (0x00007c00)
Next at t=2086087
(0) [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00
<bochs:11> s
Next at t=2086088
(0) [0x00007c3e] 0000:7c3e (unk. ctxt): cli ; fa
<bochs:12> s
Next at t=2086089
(0) [0x00007c3f] 0000:7c3f (unk. ctxt): mov ax, 0x07c0 ; b8c007
<bochs:13> s
Next at t=2086090
(0) [0x00007c42] 0000:7c42 (unk. ctxt): mov ds, ax ; 8ed8
<bochs:14> s
Next at t=2086091
(0) [0x00007c44] 0000:7c44 (unk. ctxt): mov es, ax ; 8ec0
<bochs:15> s
Next at t=2086092
(0) [0x00007c46] 0000:7c46 (unk. ctxt): mov fs, ax ; 8ee0
<bochs:16> _
```



8. Tambahkan ‘break-point’ yang lain (maksimal 7). Tahapan penting berikutnya adalah ketika PC Simulator mulai menjalankan program ‘kernel.bin’, hal ini terjadi pada alamat ‘0100:0000’.

```
Bochs for Windows - Console
rdx: 0x00000000:0000f20 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 vlp vlf ac vm rf nt of df lf tf sf zf af pf cf
<bochs:4> s
Next at t=2
(0) [0x000fe05d] f000:e05d (unk. ctxt): out 0xd, al ; e60d
<bochs:5> s
Next at t=3
(0) [0x000fe05f] f000:e05f (unk. ctxt): out 0xda, al ; e6da
<bochs:6> s
Next at t=4
(0) [0x000fe061] f000:e061 (unk. ctxt): mov al, 0xc0 ; b0c0
<bochs:7> s
Next at t=5
(0) [0x000fe063] f000:e063 (unk. ctxt): out 0xd6, al ; e6d6
<bochs:8> vb 0:0x7c00
<bochs:9> vb 0:0x7c00
<bochs:10> c
(2005679513) Breakpoint 10285608, ln 0000:7c00 (0x00007c00)
Next at t=2086087
(0) [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00
<bochs:11> s
Next at t=2086088
(0) [0x00007c3e] 0000:7c3e (unk. ctxt): cli ; fa
<bochs:12> s
Next at t=2086089
(0) [0x00007c3f] 0000:7c3f (unk. ctxt): mov ax, 0x07c0 ; b0c007
<bochs:13> s
Next at t=2086090
(0) [0x00007c42] 0000:7c42 (unk. ctxt): mov ds, ax ; 8ed8
<bochs:14> s
Next at t=2086091
(0) [0x00007c44] 0000:7c44 (unk. ctxt): mov es, ax ; 8ec0
<bochs:15> s
Next at t=2086092
(0) [0x00007c46] 0000:7c46 (unk. ctxt): mov fs, ax ; 8ee0
<bochs:16> s
Next at t=2086093
(0) [0x00007c48] 0000:7c48 (unk. ctxt): mov gs, ax ; 8ee8
<bochs:17> s
Next at t=2086094
(0) [0x00007c4a] 0000:7c4a (unk. ctxt): mov ax, 0x0000 ; b00000
<bochs:18> _
```

9. Hentikan PC Simulator pada saat akan menjalankan program 'kernel.bin':  
Mulailah dari awal, hentikan 'debugging' sebelumnya dengan memasukan perintah 'q', jika PC Simulator sedang bekerja, hentikan dengan menekan tombol 'CTRL+C' kemudian 'q'. Selanjutnya mulai dari awal, ketik 's'. Kemudian buatlah break-point, masukan perintah 'vb 0x0100:0x0000' untuk menghentikan langkah saat PC mulai mengeksekusi instruksi dari program 'kernel.bin'. Selanjutnya perintahkan PC untuk melanjutkan pekerjaan, 'c'. Jika langkah anda sesuai dengan petunjuk di atas maka pada layar 'debugging' akan di tampilkan teks berikut:



```
Bochs for Windows - Console
(2005679513) Breakpoint 10285608, in 0000:7c00 (0x00007c00)
Next at t=2086087
(0) [0x00007c00] 0000:7c00 (unk. ctxt): jmp .+0x003b (0x00007c3e) ; e93b00
<bochs:11> s
Next at t=2086088
(0) [0x00007c3e] 0000:7c3e (unk. ctxt): cli ; fa
<bochs:12> s
Next at t=2086089
(0) [0x00007c3f] 0000:7c3f (unk. ctxt): mov ax, 0x07c0 ; b8c007
<bochs:13> s
Next at t=2086090
(0) [0x00007c42] 0000:7c42 (unk. ctxt): mov ds, ax ; 8ed8
<bochs:14> s
Next at t=2086091
(0) [0x00007c44] 0000:7c44 (unk. ctxt): mov es, ax ; 8ec0
<bochs:15> s
Next at t=2086092
(0) [0x00007c46] 0000:7c46 (unk. ctxt): mov fs, ax ; 8ee0
<bochs:16> s
Next at t=2086093
(0) [0x00007c48] 0000:7c48 (unk. ctxt): mov gs, ax ; 8ee8
<bochs:17> s
Next at t=2086094
(0) [0x00007c4a] 0000:7c4a (unk. ctxt): mov ax, 0x0000 ; b80000
<bochs:18> q
# In bx_win32_gui_c::exit(void)!

Bochs is exiting. Press ENTER when you're ready to close this window.
Terminate batch job (Y/N)? Y

C:\> s
C:\> ..\..\bochs-2.3.5\bochsrc -q -f bochsrc.bxrc
000000000001[APIC?] local apic in Initializing
=====
Bochs x86 Emulator 2.3.5
Build from CVS snapshot, on September 16, 2007
=====
000000000001[ ] reading configuration from bochsrc.bxrc
000000000001[ ] installing win32 module as the Bochs GUI
000000000001[ ] using log file bochs.log
Next at t=0
(0) [0xfffffff0] 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> _
```

10. Selanjutnya teruskan langkah PC Simulator step-by-step minimal sebanyak 10x, ketik 's', step berikutnya dapat dilakukan dengan cara menekan tombol secara langsung.

```

Bochs for Windows - Console
..\\..\\bochs-2.3.5\\bochsrc -q -f bochsrc.bxrc
000000000000[APIC?] local apic in initializing
=====
Bochs x86 Emulator 2.3.5
Build from CVS snapshot, on September 16, 2007
=====
000000000000[ ] reading configuration from bochsrc.bxrc
000000000000[ ] installing win32 module as the Bochs GUI
000000000000[ ] 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 10265608, 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): cll ; 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): stl ; 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>
Next at t=2945024
(0) [0x00001014] 0100:0014 (unk. ctxt): cll ; fa
<bochs:14> _

```

11. Langkah-langkah di atas adalah merupakan salah satu cara yang banyak di gunakan para pengembang perangkat lunak, termasuk pengembangan sistim operasi.

## TUGAS

1. Tabel pemetaan memori pada PC lengkap

No	Jenis Memori	Nama Memori	Deskripsi Memori
1	Memori Fisik	RAM (Random Access Memory)	Tempat utama untuk data dan program aktif
2	Memori Cache	Cache Level 1 (L1)	Cache tercepat di dalam CPU
3		Cache Level 2 (L2)	Cache di dalam CPU dengan kapasitas lebih besar
4		Cache Level 3 (L3)	Cache yang lebih besar dengan kecepatan lebih rendah
5	Memori Virtual	File Paging	Penggunaan hard drive sebagai memori virtual
6		Swap Space (Swap Partition)	Ruang pada hard drive untuk fungsi swap

7	Memori Grafis	VRAM (Video RAM)	Memori untuk kartu grafis dan rendering
8	BIOS/UEFI Firmware	BIOS (Basic Input/Output System) atau UEFI (Unified Extensible Firmware Interface)	Program firmware untuk inisialisasi dan booting
9	Memori	Kartu suara (Sound Card)	Memori untuk pengolahan audio
10	Perangkat Tambahan	Kartu jaringan (Network Card)	Memori untuk pengolahan data jaringan
11	I/O Memori	I/O Ports	Memori untuk mengendalikan perangkat I/O
12		Memory-Mapped I/O	
13	Memori Sistem (System Memory)	Low Memory (0-640KB)	Area kuno untuk BIOS dan driver perangkat keras
14		Upper Memory (640KB-1MB)	Area untuk penggunaan khusus DOS
15		Extended Memory (Lebih dari 1MB)	Memori konvensional digunakan oleh sistem operasi

## 2. Perbedaan antara mode kerja ‘Real-Mode’ dan mode kerja ‘Protect-Mode’ pada PC IBM Compatible

### a. Akses Memori

- Dalam mode Real-Mode, prosesor memiliki akses langsung ke seluruh alamat memori fisik komputer. Ini berarti bahwa program dalam mode Real-Mode dapat mengakses semua area memori tanpa pembatasan, yang dapat menyebabkan masalah jika program salah mengakses atau merusak data atau kode lainnya.
- Dalam mode Protect-Mode, sistem mengimplementasikan kontrol akses memori yang lebih ketat. Ini memungkinkan sistem operasi untuk mengisolasi program-program yang berjalan, membatasi akses mereka ke wilayah memori tertentu, dan mencegah program satu dari merusak program atau data lainnya.

### b. Perlindungan dan Keamanan

- Mode Real-Mode tidak memiliki mekanisme bawaan untuk melindungi program dari tindakan yang merusak. Semua perangkat lunak berjalan dalam mode Real-Mode memiliki akses penuh ke semua sumber daya sistem.
- Protect-Mode mengaktifkan kontrol yang lebih baik atas akses ke sumber daya sistem, sehingga memungkinkan sistem operasi untuk memberikan izin dan batasan kepada program sesuai dengan tingkat hak akses yang telah ditetapkan.

### c. Kapasitas Memori

- Mode Real-Mode hanya dapat mengakses 1 MB memori fisik. Ini adalah batasan yang signifikan pada sistem modern yang memiliki jauh lebih banyak memori fisik.
- Mode Protect-Mode dapat mengakses lebih dari 1 MB memori fisik. Ini memungkinkan sistem untuk menggunakan memori yang lebih besar dan lebih efisien.

### d. Penggunaan Sumber Daya

- Mode Real-Mode kurang efisien dalam penggunaan sumber daya karena program tidak terisolasi satu sama lain, sehingga dapat mengganggu kinerja dan kestabilan sistem.

- Mode Protect-Mode memberikan isolasi yang lebih baik antara program, sehingga lebih efisien dalam penggunaan sumber daya dan meningkatkan stabilitas sistem.
- e. Kompatibilitas
- Mode Real-Mode diperlukan untuk menjalankan program yang dikembangkan untuk arsitektur prosesor x86 kuno, seperti program MS-DOS.
  - Mode Protect-Mode diperlukan untuk menjalankan sistem operasi modern seperti Windows, Linux, dan macOS, serta aplikasi yang lebih kompleks.