



MODUL 1

Pengenalan Sistem Pengembangan OS dengan PC Simulator 'Bochs'

Penyusun:

Nama : Della Fitria Lestari
NIM : L200219268
Kelas : E
Mata Kuliah : Praktikum Sistem Operasi
Program Studi : Informatika

ILMU KOMUNIKASI & INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
2022

Tugas

1. Kode ASCII (American Standard Code for Information Interchange)
- 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.

Desimal	Heksadesimal	Binary	Simbol/Char	Keterangan
0	00	00000000	NUL	Null char
1	01	00000001	SOH	Start of Heading
2	02	00000010	STX	Start of Text
3	03	00000011	ETX	End of Text
4	04	00000100	EOT	End of Transmission
5	05	00000101	ENQ	Enquiry
6	06	00000110	ACK	Acknowledgment
7	07	00000111	BEL	Bell
8	08	00001000	BS	Back Space
9	09	00001001	HT	Horizontal Tab
10	0A	00001010	LF	Line Feed
11	0B	00001011	VT	Vertical Tab
12	0C	00001100	FF	Form Feed
13	0D	00001101	CR	Carriage Return
14	0E	00001110	SO	Shift Out / X-On
15	0F	00001111	SI	Shift In / X-Off
16	10	00010000	DLE	Data Line Escape
17	11	00010001	DC1	Device Control 1 (oft. XON)
18	12	00010010	DC2	Device Control 2
19	13	00010011	DC3	Device Control 3 (oft. XOFF)
20	14	00010100	DC4	Device Control 4
21	15	00010101	NAK	Negative Acknowledgement
22	16	00010110	SYN	Synchronous Idle
23	17	00010111	ETB	End of Transmit Block
24	18	00011000	CAN	Cancel
25	19	00011001	EM	End of Medium
26	1A	00011010	SUB	Substitute
27	1B	00011011	ESC	Escape
28	1C	00011100	FS	File Separator
29	1D	00011101	GS	Group Separator
30	1E	00011110	RS	Record Separator
31	1F	00011111	US	Unit Separator
32	20	00100000		Space
33	21	00100001	!	Exclamation mark
34	22	00100010	"	Double quotes (or speech marks)
35	23	00100011	#	Number
36	24	00100100	\$	Dollar
37	25	00100101	%	Procenttecken
38	26	00100110	&	Ampersand
39	27	00100111	'	Single quote
40	28	00101000	(Open parenthesis (or

				open bracket)
41	29	00101001)	Close parenthesis (or close bracket)
42	2A	00101010	*	Asterisk
43	2B	00101011	+	Plus
44	2C	00101100	,	Comma
45	2D	00101101	-	Hyphen
46	2E	00101110	.	Period, dot or full stop
47	2F	00101111	/	Slash or divide
48	30	00110000	0	Zero
49	31	00110001	1	One
50	32	00110010	2	Two
51	33	00110011	3	Three
52	34	00110100	4	Four
53	35	00110101	5	Five
54	36	00110110	6	Six
55	37	00110111	7	Seven
56	38	00111000	8	Eight
57	39	00111001	9	Nine
58	3A	00111010	:	Colon
59	3B	00111011	;	Semicolon
60	3C	00111100	<	Less than (or open angled bracket)
61	3D	00111101	=	Equals
62	3E	00111110	>	Greater than (or close angled bracket)
63	3F	00111111	?	Question mark
64	40	01000000	@	At symbol
65	41	01000001	A	Uppercase A
66	42	01000010	B	Uppercase B
67	43	01000011	C	Uppercase C
68	44	01000100	D	Uppercase D
69	45	01000101	E	Uppercase E
70	46	01000110	F	Uppercase F
71	47	01000111	G	Uppercase G
72	48	01001000	H	Uppercase H
73	49	01001001	I	Uppercase I
74	4A	01001010	J	Uppercase J
75	4B	01001011	K	Uppercase K
76	4C	01001100	L	Uppercase L
77	4D	01001101	M	Uppercase M
78	4E	01001110	N	Uppercase N
79	4F	01001111	O	Uppercase O
80	50	01010000	P	Uppercase P
81	51	01010001	Q	Uppercase Q
82	52	01010010	R	Uppercase R
83	53	01010011	S	Uppercase S
84	54	01010100	T	Uppercase T
85	55	01010101	U	Uppercase U
86	56	01010110	V	Uppercase V
87	57	01010111	W	Uppercase W
88	58	01011000	X	Uppercase X

89	59	01011001	Y	Uppercase Y
90	5A	01011010	Z	Uppercase Z
91	5B	01011011	[Opening bracket
92	5C	01011100	\	Backslash
93	5D	01011101]	Closing bracket
94	5E	01011110	^	Caret - circumflex
95	5F	01011111	_	Underscore
96	60	01100000	`	Grave accent
97	61	01100001	a	Lowercase a
98	62	01100010	b	Lowercase b
99	63	01100011	c	Lowercase c
100	64	01100100	d	Lowercase d
101	65	01100101	e	Lowercase e
102	66	01100110	f	Lowercase f
103	67	01100111	g	Lowercase g
104	68	01101000	h	Lowercase h
105	69	01101001	i	Lowercase i
106	6A	01101010	j	Lowercase j
107	6B	01101011	k	Lowercase k
108	6C	01101100	l	Lowercase l
109	6D	01101101	m	Lowercase m
110	6E	01101110	n	Lowercase n
111	6F	01101111	o	Lowercase o
112	70	01110000	p	Lowercase p
113	71	01110001	q	Lowercase q
114	72	01110010	r	Lowercase r
115	73	01110011	s	Lowercase s
116	74	01110100	t	Lowercase t
117	75	01110101	u	Lowercase u
118	76	01110110	v	Lowercase v
119	77	01110111	w	Lowercase w
120	78	01111000	x	Lowercase x
121	79	01111001	y	Lowercase y
122	7A	01111010	z	Lowercase z
123	7B	01111011	{	Opening brace
124	7C	01111100		Vertical bar
125	7D	01111101	}	Closing brace
126	7E	01111110	~	Equivalency sign - tilde
127	7F	01111111		Delete
128	80	10000000	€	Euro sign
129	81	10000001		
130	82	10000010	,	Single low-9 quotation mark
131	83	10000011	<i>f</i>	Latin small letter f with hook
132	84	10000100	„	Double low-9 quotation mark
133	85	10000101	...	Horizontal ellipsis
134	86	10000110	†	Dagger
135	87	10000111	‡	Double dagger
136	88	10001000	ˆ	Modifier letter circumflex accent
137	89	10001001	‰	Per mille sign

138	8A	10001010	Š	Latin capital letter S with caron
139	8B	10001011	◁	Single left-pointing angle quotation
140	8C	10001100	Œ	Latin capital ligature OE
141	8D	10001101		
142	8E	10001110	Ž	Latin captial letter Z with caron
143	8F	10001111		
144	90	10010000		
145	91	10010001	‘	Left single quotation mark
146	92	10010010	’	Right single quotation mark
147	93	10010011	“	Left double quotation mark
148	94	10010100	”	Right double quotation mark
149	95	10010101	•	Bullet
150	96	10010110	—	En dash
151	97	10010111	—	Em dash
152	98	10011000	˜	Small tilde
153	99	10011001	™	Trade mark sign
154	9A	10011010	š	Latin small letter S with caron
155	9B	10011011	›	Single right-pointing angle quotationmark
156	9C	10011100	œ	Latin small ligature oe
157	9D	10011101		
158	9E	10011110	ž	Latin small letter z with caron
159	9F	10011111	Ÿ	Latin capital letter Y with diaeresis
160	A0	10100000		Non-breaking space
161	A1	10100001	¡	Inverted exclamation mark
162	A2	10100010	¢	Cent sign
163	A3	10100011	£	Pound sign
164	A4	10100100	¤	Currency sign
165	A5	10100101	¥	Yen sign
166	A6	10100110		Pipe, Broken vertical bar
167	A7	10100111	§	Section sign
168	A8	10101000	¨	Spacing diaeresis - umlaut
169	A9	10101001	©	Copyright sign
170	AA	10101010	ª	Feminine ordinal indicator
171	AB	10101011	«	Left double angle quotes
172	AC	10101100	¬	Not sign
173	AD	10101101		Soft hyphen
174	AE	10101110	®	Registered trade mark sign
175	AF	10101111	-	Spacing macron - overline

176	B0	10110000	°	Degree sign
177	B1	10110001	±	Plus-or-minus sign
178	B2	10110010	²	Superscript two - squared
179	B3	10110011	³	Superscript three - cubed
180	B4	10110100	´	Acute accent - spacing acute
181	B5	10110101	μ	Micro sign
182	B6	10110110	¶	Pilcrow sign - paragraph sign
183	B7	10110111	.	Middle dot - Georgian comma
184	B8	10111000	¸	Spacing cedilla
185	B9	10111001	¹	Superscript one
186	BA	10111010	º	Masculine ordinal indicator
187	BB	10111011	»	Right double angle quotes
188	BC	10111100	¼	Fraction one quarter
189	BD	10111101	½	Fraction one half
190	BE	10111110	¾	Fraction three quarters
191	BF	10111111	¿	Inverted question mark
192	C0	11000000	À	Latin capital letter A with grave
193	C1	11000001	Á	Latin capital letter A with acute
194	C2	11000010	Â	Latin capital letter A with circumflex
195	C3	11000011	Ã	Latin capital letter A with tilde
196	C4	11000100	Ä	Latin capital letter A with diaeresis
197	C5	11000101	Å	Latin capital letter A with ring above
198	C6	11000110	Æ	Latin capital letter AE
199	C7	11000111	Ç	Latin capital letter C with cedilla
200	C8	11001000	È	Latin capital letter E with grave
201	C9	11001001	É	Latin capital letter E with acute
202	CA	11001010	Ê	Latin capital letter E with circumflex
203	CB	11001011	Ë	Latin capital letter E with diaeresis
204	CC	11001100	Ì	Latin capital letter I with grave
205	CD	11001101	Í	Latin capital letter I with acute
206	CE	11001110	Î	Latin capital letter I with circumflex
207	CF	11001111	Ï	Latin capital letter I with diaeresis
208	D0	11010000	Ð	Latin capital letter ETH
209	D1	11010001	Ñ	Latin capital letter N

				with tilde
210	D2	11010010	Ò	Latin capital letter O with grave
211	D3	11010011	Ó	Latin capital letter O with acute
212	D4	11010100	Ô	Latin capital letter O with circumflex
213	D5	11010101	Õ	Latin capital letter O with tilde
214	D6	11010110	Ö	Latin capital letter O with diaeresis
215	D7	11010111	×	Multiplication sign
216	D8	11011000	Ø	Latin capital letter O with slash
217	D9	11011001	Ù	Latin capital letter U with grave
218	DA	11011010	Ú	Latin capital letter U with acute
219	DB	11011011	Û	Latin capital letter U with circumflex
220	DC	11011100	Ü	Latin capital letter U with diaeresis
221	DD	11011101	Ý	Latin capital letter Y with acute
222	DE	11011110	Þ	Latin capital letter THORN
223	DF	11011111	ß	Latin small letter sharp s - ess-zed
224	E0	11100000	à	Latin small letter a with grave
225	E1	11100001	á	Latin small letter a with acute
226	E2	11100010	â	Latin small letter a with circumflex
227	E3	11100011	ã	Latin small letter a with tilde
228	E4	11100100	ä	Latin small letter a with diaeresis
229	E5	11100101	å	Latin small letter a with ring above
230	E6	11100110	æ	Latin small letter ae
231	E7	11100111	ç	Latin small letter c with cedilla
232	E8	11101000	è	Latin small letter e with grave
233	E9	11101001	é	Latin small letter e with acute
234	EA	11101010	ê	Latin small letter e with circumflex
235	EB	11101011	ë	Latin small letter e with diaeresis
236	EC	11101100	ì	Latin small letter i with grave
237	ED	11101101	í	Latin small letter i with acute
238	EE	11101110	î	Latin small letter i with

				circumflex
239	EF	11101111	ï	Latin small letter i with diaeresis
240	F0	11110000	ð	Latin small letter eth
241	F1	11110001	ñ	Latin small letter n with tilde
242	F2	11110010	ò	Latin small letter o with grave
243	F3	11110011	ó	Latin small letter o with acute
244	F4	11110100	ô	Latin small letter o with circumflex
245	F5	11110101	õ	Latin small letter o with tilde
246	F6	11110110	ö	Latin small letter o with diaeresis
247	F7	11110111	÷	Division sign
248	F8	11111000	ø	Latin small letter o with slash
249	F9	11111001	ù	Latin small letter u with grave
250	FA	11111010	ú	Latin small letter u with acute
251	FB	11111011	û	Latin small letter u with circumflex
252	FC	11111100	ü	Latin small letter u with diaeresis
253	FD	11111101	ý	Latin small letter y with acute
254	FE	11111110	þ	Latin small letter thorn
255	FF	11111111	ÿ	Latin small letter y with diaeresis

2. Daftar perintah bahasa assembly untuk mesin intel keluarga x86 lengkap
- Daftar Instruksi (yaitu kode yang harus dieksekusi oleh CPU mikrokontroler dengan mela tertentu sesuai dengan daftar yang sudah tertanam dalam CPU)

No.	Instruksi	Fungsi
1.	ACALL (Absolute Call)	memanggil sub rutin program
2.	ADD (Add Immediate Data)	menambah 8 bit data langsung ke dalam isi akumulator dan menyimpan hasilnya pada akumulator
3.	ADDC (Add Carry Plus Immediate Data to Accumulator)	menambahkan isi carry flag (0 atau 1) ke dalam isi akumulator. Data langsung 8 bit ditambahkan ke akumulator.
4.	AJMP (Absolute Jump)	mentransfer kendali program ke lokasi dimana alamat dikalkulasi dengan cara yang sama dengan perintah ACALL.
5.	ANL (logical AND memori ke akumulator)	mengAND-kan isi alamat data dengan isi akumulator.
6.	CJNE (Compare Indirect Address to Immediate Data)	membandingkan data langsung dengan lokasi memori yang dialamati oleh register R atau Akumulator A. apabila tidak sama maka instruksi akan menuju ke alamat kode. Format : CJNE R,#data,Alamat kode.

7.	CLR (Clear Accumulator)	mereset data akumulator menjadi 00H. Format : CLR A
8.	CPL (Complement Accumulator)	mengkomplemen isi akumulator.
9.	DA (Decimal Adjust Accumulator)	mengatur isi akumulator ke padanan BCD, steleah penambahan dua angka BCD.
10.	DEC (Decrement Indirect Address)	mengurangi isi lokasi memori yang ditujukan oleh register R dengan 1, dan hasilnya disimpan pada lokasi tersebut.
11.	DIV (Divide Accumulator by B)	membagi isi akumulator dengan isi register B. Akumulator berisi hasil bagi, register B berisi sisa pembagian.
12.	DJNZ (Decrement Register And Jump Id Not Zero)	mengurangi nilai register dengan 1 dan jika hasilnya sudah 0 maka instruksi selanjutnya akan dieksekusi. Jika belum 0 akan menuju ke alamat kode.
13.	INC (Increment Indirect Address)	menambahkan isi memori dengan 1 dan menyimpannya pada alamat tersebut.
14.	JB (Jump if Bit is Set)	membaca data per satu bit, jika data tersebut adalah 1 maka akan menuju ke alamat kode dan jika 0 tidak akan menuju ke alamat kode.
15.	JBC (Jump if Bit Set and Clear Bit)	perintah rel menguji yang terspesifikasikan secara bit. Jika bit di-set, maka Jump dilakukan ke alamat relatif dan yang terspesifikasi secara bit di dalam perintah dibersihkan. Segmen program berikut menguji bit yang kurang signifikan (LSB: Least Significant Byte), dan jika diketemukan bahwa ia telah di-set, program melompat ke READ lokasi. JBC juga berfungsi membersihkan LSB dari akumulator.
16.	JC (Jump if Carry is Set)	menguji isi carry flag. Jika berisi 1, eksekusi menuju ke alamat kode, jika berisi 0, instruksi selanjutnya yang akan dieksekusi.
17.	JMP (Jump to sum of Accumulator and Data Pointer)	memerintah loncat kesuato alamat kode tertentu. Format : JMP alamat kode.
18.	JNB (Jump if Bit is Not Set)	membaca data per satu bit, jika data tersebut adalah 0 maka akan menuju ke alamat kode dan jika 1 tidak akan menuju ke alamat kode. Format : JNB alamat bit,alamat kode.
19.	JNC (Jump if Carry Not Set)	menguji bit Carry, dan jika tidak di-set, maka sebuah lompatan akan dilakukan ke alamat relatif yang telah ditentukan.
20.	JNZ (Jump if Accumulator Not Zero)	Jump if not zero (lompat jika tidak nol). Dalam hal ini suatu lompatan akan terjadi bilamana bendera nol dalam keadaan “clear”, dan tidak akan terjadi lompatan bilamana bendera nol tersebut dalam keadaan set. Andaikan bahwa JNZ 7800H disimpan pada lokasi 2100H. Jika Z=0, instruksi berikutnya akan berasal dari lokasi 7800H: dan bilamana Z=1, program akan turun ke instruksi urutan berikutnya pada lokasi 2101H.
21.	JZ (Jump if Accumulator is Zero)	menguji konten-konten akumulator. Jika bukan nol, maka lompatan dilakukan ke alamat relatif yang ditentukan dalam perintah.

22.	LCALL (Long Call)	memungkinkan panggilan ke subrutin yang berlokasi dimanapun dalam memori program 64K. Operasi LCALL berjalan seperti berikut: <ul style="list-style-type: none"> · Menambahkan ke dalam konter program sebanyak 3, karena perintahnya adalah perintah 3-byte. · Menambahkan penunjuk stack sebanyak 1. · Menyimpan byte yang lebih rendah dari konter program ke dalam stack. · Menambahkan penunjuk stack. · Menyimpan byte yang lebih tinggi dari program ke dalam stack. · Me-load konter program dengan alamat tujuan 16-bit.
23.	LJMP (Long Jump)	memungkinkan lompatan tak bersyarat kemana saja dalam lingkup ruang memori program 64K. LCALL adalah perintah 3-byte. Alamat tujuan 16-bit ditentukan secara langsung dalam perintah tersebut. Alamat tujuan ini di-load ke dalam konter program oleh perintah LJMP.
24.	MOV (Move From Memory)	memindahkan isi akumulator/register atau data dari nilai luar atau alamat lain.
25.	MOVC (Move From Codec Memory)	mengisi accumulator dengan byte kode atau konstanta dari program memory. Alamat byte tersebut adalah hasil penjumlahan unsigned 8 bit pada accumulator dan 16 bit register basis yang dapat berupa data pointer atau program counter. Instruksi ini tidak mempengaruhi flag apapun juga.
26.	MOVX (Move Accumulator to External Memory Addressed by Data Pointer)	memindahkan isi akumulator ke memori data eksternal yang alamatnya ditunjukkan oleh isi data pointer.
27.	MUL (Multiply)	mengalikan unsigned 8 bit integer pada accumulator dan register B.
28.	NOP (No Operation)	eksekusi program akan dilanjutkan ke instruksi berikutnya.
29.	ORL (Logical OR Immediate Data to Accumulator)	sebagai instruksi Gerbang logika OR yang akan menjumlahkan Accumulator terhadap nilai yang ditentukan. Format : ORL A,#data.
30.	POP (Pop Stack to Memory)	menempatkan byte yang ditunjukkan oleh stack pointer ke suatu alamat data.
31.	PUSH (Push Memory onto Stack)	menaikkan stack pointer kemudian menyimpan isinya ke suatu alamat data pada lokasi yang ditunjuk oleh stack pointer.
32.	RET (Return from subroutine)	untuk kembali dari suatu subrutin program ke alamat terakhir subrutin tersebut di panggil.
33.	RETI (Return From Interrupt)	mengambil nilai byte tinggi dan rendah dari PC dari stack dan mengembalikan kondisi logika interrupt agar dapat menerima interrupt lain dengan prioritas yang sama dengan prioritas interrupt yang baru saja diproses.
34.	RL (Rotate Accumulator Left)	memutar setiap bit dalam akumulator satu posisi ke kiri.
35.	RLC (Rotate Left through Carry)	memutar (Rotate) Accumulator ke Kiri (Left) Melalui Carry Flag. Kedelapan bit accumulator dan carry flag akan diputar satu bit ke kiri secara bersama-sama.

36.	RR (Rotate Right)	memutar (Rotate) Accumulator ke Kanan (Right). Kedelapan bit accumulator akan diputar satu bit ke kanan.
37.	RRC (Rotate Right through Carry)	Memutar (Rotate) Accumulator ke Kanan (Right) Melalui Carry Flag. Kedelapan bit accumulator dan carry flag akan diputar satu bit ke kanan secara bersama-sama.
38.	SETB (set Carry flag)	untuk menset carry flag
39.	SJMP (Short Jump)	mentransfer kendali ke alamat tujuan dalam 127 bytes yang mengikuti dan 128 yang mengawali perintah SJMP.
40.	SUBB (Subtract With Borrow)	Pengurangan (Subtract) dengan Peminjaman (Borrow).
41.	SWAP (Swap Nibbles)	Menukar (Swap) Upper Nibble dan Lower Nibble dalam Accumulator.
42.	XCH (Exchange Bytes)	Menukar (Exchange) Accumulator dengan Variabel Byte.
43.	XCHD (Exchange Digits)	Menukar (Exchange) Digit. XCHD menukar nibble rendah dari accumulator, yang umumnya mewakili angka heksadesimal atau BCD, dengan nibble rendah dari internal data memory yang diakses secara indirect.
44.	XRL (Exclusive OR Logic)	Logika Exclusive OR untuk Variabel Byte XRL akan melakukan operasi bitwise logika exclusive OR antara kedua variabel yang dinyatakan.

Daftar Assembly Directive (yaitu merupakan kode yang menjadi arahan bagi assembler/ menata program)

No.	Assembly directive	Keterangan
1.	EQU	Pendefinisian konstanta
2.	DB	Pendefinisian data dengan ukuran satuan 1 byte
3.	DW	Pendefinisian data dengan ukuran satuan 1 word
4.	DBIT	Pendefinisian data dengan ukuran satuan 1 bit
5.	DS	Pemesanan tempat penyimpanan data di RAM
6.	ORG	Inisialisasi alamat mulai program
7.	END	Penanda akhir program
8.	CSEG	Penanda penempatan di code segment
9.	XSEG	Penanda penempatan di external data segment
10.	DSEG	Penanda penempatan di internal direct data segment
11.	ISEG	Penanda penempatan di internal indirect data segment
12.	BSEG	Penanda penempatan di bit data segment
13.	CODE	Penanda mulai pendefinisian program
14.	XDATA	Pendefinisian external data
15.	DATA	Pendefinisian internal direct data
16.	IDATA	Pendefinisian internal indirect data
17.	BIT	Pendefinisian data bit
18.	#INCLUDE	Mengikutsertakan file program lain

Langkah Kerja

1. Menjalankan program command prompt atau cmd.
2. Masukkan perintah cd C:\OS untuk masuk ke direktori kerja C:\OS
3. Untuk melihat isi dari direktori masukan perintah dir. Seperti dibawah:

```
CA: Command Prompt
Microsoft Windows [Version 10.0.22000.856]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>cd\

C:\>cd os

C:\OS>dir
Volume in drive C is OS
Volume Serial Number is 8E24-6DC9

Directory of C:\OS

17/09/2019  16:47    <DIR>          .
17/09/2019  16:47    <DIR>          Bochs-2.3.5
17/09/2019  16:47    <DIR>          Dev-Cpp
17/12/2008  00:08             1.096.291  i386.pdf
17/09/2019  16:47    <DIR>          LAB
17/12/2008  00:07             846.920  pcasm-book.pdf
17/12/2008  01:44              86  Setpath.bat
13/12/2008  14:12             716.512  winima81.exe
               4 File(s)          2.659.809 bytes
               4 Dir(s)  33.116.585.984 bytes free

C:\OS>
```

4. Menjalankan file setpath, untuk menjalankan ketik ‘setpath’ tekan <ENTER>

```
CA: Command Prompt
C:\OS>setpath

C:\OS>Path=C:\OS\Dev-Cpp\bin;C:\OS\Bochs-2.3.5;c:\OS\Per1;C:\Windows;C:\Windows\System32
C:\OS>
```

5. Masuk ke dikrektori kerja pada ‘C:\OS\LAB\LAB1’ dengan cara cd lab lalu <ENTER> cd lab1 <ENTER> dan buka isi direktori dengan dir. Seperti gambar dibawah:

```
CA: Bochs for Windows - Console
C:\OS>cd lab

C:\OS\LAB>cd lab1

C:\OS\LAB\LAB1>dir
Volume in drive C is OS
Volume Serial Number is 8E24-6DC9

Directory of C:\OS\LAB\LAB1

15/09/2022  16:40    <DIR>          .
17/09/2019  16:47    <DIR>          ..
15/09/2022  16:50             10.217  bochsout.txt
15/12/2008  16:17             1.628  bochsrc.bxrc
14/12/2008  12:02             14.365  boot.asm
15/09/2022  16:39              512  boot.bin
16/09/2015  07:51              512  boots.bin
15/12/2008  00:47              78  dosfp.bat
15/09/2022  16:48             1.474.560  floppya.img
14/12/2008  11:45              7.966  kernel.asm
15/12/2008  16:21              227  Makefile
15/12/2008  12:20              44  s.bat
               10 File(s)          1.510.109 bytes
                2 Dir(s)  33.115.131.904 bytes free
```

6. Membuka file dengan perintah, ketik ‘Notepad boot.asm’ dan tekan <ENTER> kemudian melihat isi direktori dan setelah itu mengetik ‘notepad makefile’. Seperti gambar dibawah ini:

Command Prompt

C:\OS\LAB\LAB1>notepad boot.asm

C:\OS\LAB\LAB1>dir
Volume in drive C is OS
Volume Serial Number is 8E24-6DC9

Directory of C:\OS\LAB\LAB1

15/09/2022 16:40 <DIR> .
17/09/2019 16:47 <DIR> ..
15/09/2022 16:50 10.217 bochsout.txt
15/12/2008 16:17 1.628 bochsrc.bxrc
14/12/2008 12:02 14.365 boot.asm
15/09/2022 16:39 512 boot.bin
16/09/2015 07:51 512 boots.bin
15/12/2008 00:47 78 dosfp.bat
15/09/2022 16:48 1.474.560 floppy.aimg
14/12/2008 11:45 7.966 kernel.asm
15/12/2008 16:21 227 Makefile
15/12/2008 12:20 44 s.bat
10 File(s) 1.510.109 bytes
2 Dir(s) 33.122.295.808 bytes free

C:\OS\LAB\LAB1>

boot.asm - Notepad

File Edit Format View Help
; *****
; LAB-1 : boot-strap loader - real mode
; untuk memindahkan file OS dari floppy disk format DOS FAT12
; *****

; atur mode kerja 16 bit (real-mode)
[BITS 16]

; Menentukan lokasi awal dari program
[ORG 0x0000]

; loncat ke label START
jmp START

; Keterangan format floppy disk format FAT12

OEM_ID db "QUASI-OS"
BytesPerSector dw 0x0200
SectorsPerCluster db 0x01

Command Prompt

C:\OS\LAB\LAB1>notepad makefile

C:\OS\LAB\LAB1>

Makefile - Notepad

File Edit Format View Help

LAB01 - Makefile

fp.disk: boot
dd if=boot.bin of=floppya.aimg

7.
- Mengetik make fp.disk untuk melakukan kompilasi terhadap source kode program ‘boot.asm’ sebagai outputnya file ‘boot.bin’ dan isinya disalin ke dalam bootsector file image floppya.aimg. Seperti gambar dibawah ini:

Command Prompt

C:\OS\LAB\LAB1>make fp.disk
nasm boot.asm -o boot.bin -f bin
dd if=boot.bin of=floppya.aimg
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>dir
Volume in drive C is OS
Volume Serial Number is 8E24-6DC9

Directory of C:\OS\LAB\LAB1

15/09/2022 16:40 <DIR> .
17/09/2019 16:47 <DIR> ..
15/09/2022 16:50 10.217 bochsout.txt
15/12/2008 16:17 1.628 bochsrc.bxrc
14/12/2008 12:02 14.365 boot.asm
17/09/2022 14:03 512 boot.bin
16/09/2015 07:51 512 boots.bin
15/12/2008 00:47 78 dosfp.bat
17/09/2022 14:04 1.474.560 floppy.aimg
14/12/2008 11:45 7.966 kernel.asm
15/12/2008 16:21 227 Makefile
15/12/2008 12:20 44 s.bat
10 File(s) 1.510.109 bytes
2 Dir(s) 33.117.392.896 bytes free

C:\OS\LAB\LAB1>

8.
- Menghapus file ‘floppya.aimg’ jika sudah ada pada direktori kerja anda, dari ‘Command Promt’ (lakukan dari direktori kerja) ketik ‘del floppya.aimg’ lanjutkan dengan tekan Y dan <ENTER>. Pastikan bahwa file sudah benar benar terhapus dengan perintah ‘dir’.
Selanjutnya panggil ‘bximage’ sehingga ditampilkan window seperti pada gambar berikut:

```

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

C:\OS\LAB\LAB1>dir
Volume in drive C is OS
Volume Serial Number is 8E24-6DC9

Directory of C:\OS\LAB\LAB1

17/09/2022  14:07    <DIR>          .
17/09/2019  16:47    <DIR>          ..
15/09/2022  16:50             10.217 bochsout.txt
15/12/2008  16:17             1.628 bochsrc.bxrc
14/12/2008  12:02             14.365 boot.asm
17/09/2022  14:03              512 boot.bin
16/09/2015  07:51              512 boots.bin
15/12/2008  00:47              78 dosfp.bat
14/12/2008  11:45             7.966 kernel.asm
15/12/2008  16:21             227 Makefile
15/12/2008  12:20              44 s.bat
               9 File(s)            35.549 bytes
               2 Dir(s)   33.117.413.376 bytes free

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

```

Command Prompt
C:\OS\LAB\LAB1>bxiimage
=====
                        bxiimage
                        Disk Image Creation Tool for Bochs
                        $Id: bxiimage.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]
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>_
```

```

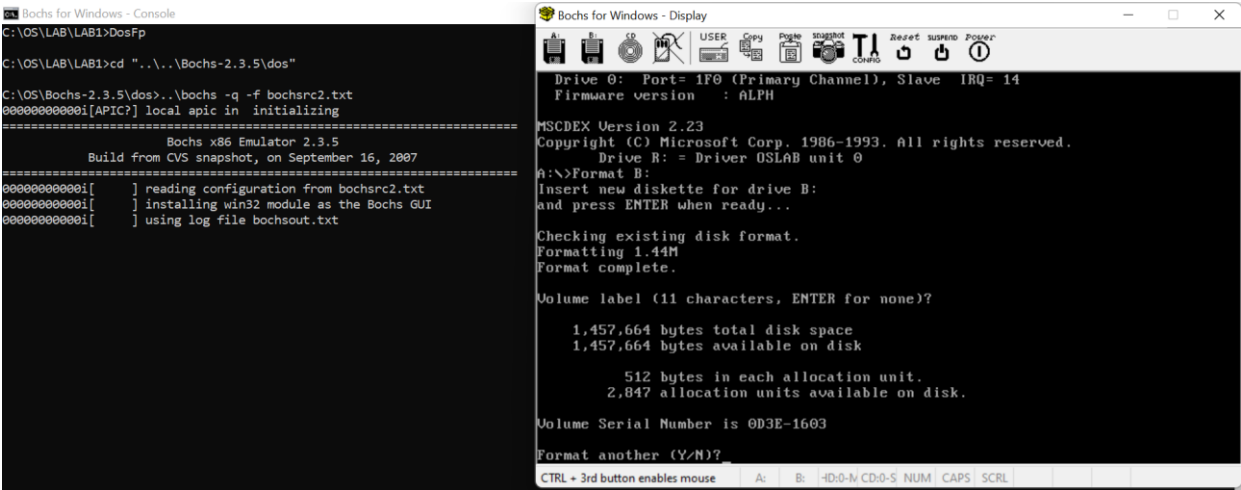
Command Prompt
C:\OS\LAB\LAB1>dir
Volume in drive C is OS
Volume Serial Number is 8E24-6DC9

Directory of C:\OS\LAB\LAB1

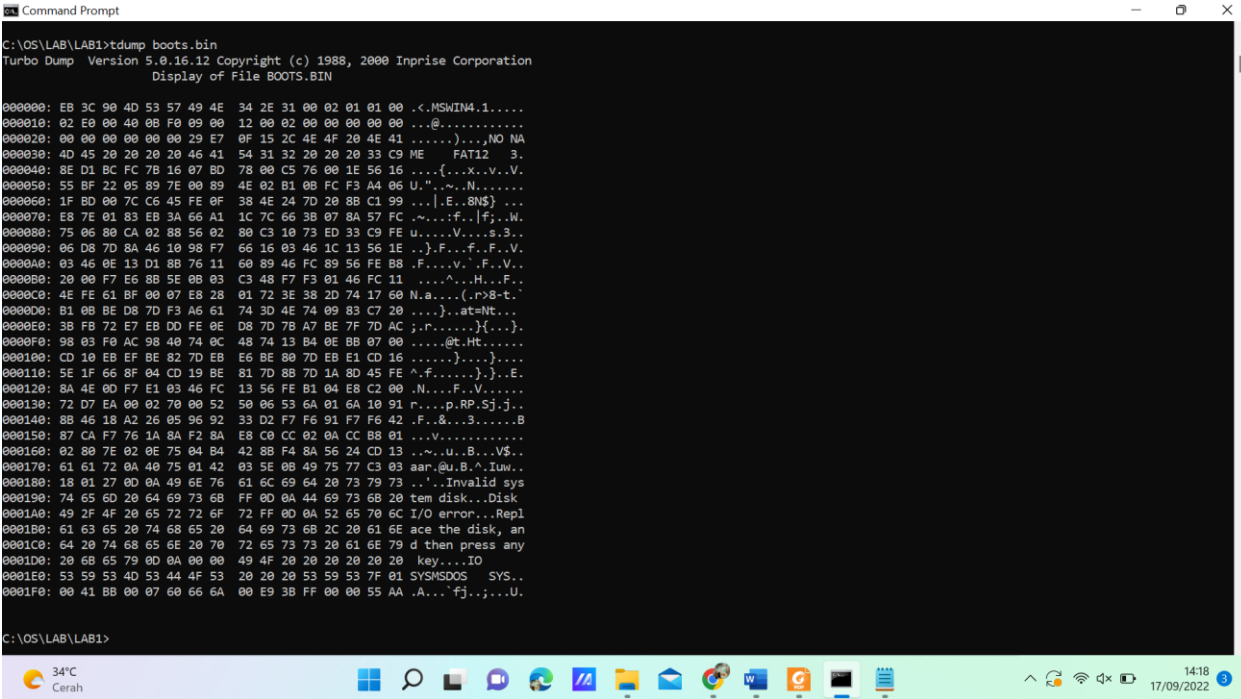
17/09/2022  14:10    <DIR>          .
17/09/2019  16:47    <DIR>          ..
17/09/2022  14:09             1.474.560 a.img
15/09/2022  16:50             10.217 bochsout.txt
15/12/2008  16:17             1.628 bochsrc.bxrc
14/12/2008  12:02             14.365 boot.asm
17/09/2022  14:03              512 boot.bin
16/09/2015  07:51              512 boots.bin
15/12/2008  00:47              78 dosfp.bat
17/09/2022  14:10             1.474.560 floppy.img
14/12/2008  11:45             7.966 kernel.asm
15/12/2008  16:21             227 Makefile
15/12/2008  12:20              44 s.bat
               11 File(s)           2.984.669 bytes
               2 Dir(s)   33.111.457.792 bytes free

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

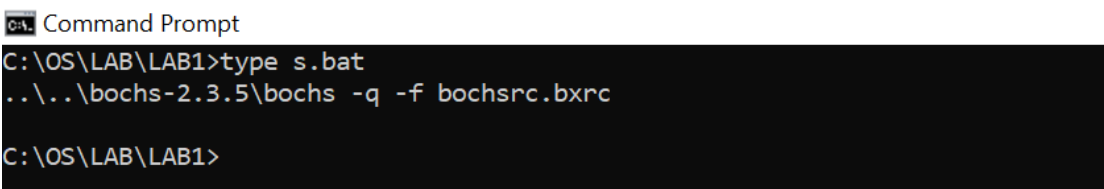
9. Menjalankan PC-Simulator dari Command Pront dengan perintah ‘DosFp’, pada konfigurasi PCSimulator file ‘floppya.img’ terpasang pada ‘drive B:’. Selanjutnya dari promp‘A:>’ ketik ‘Format B:’<ENTER> [2X], seperti ditampilkan pada gambar berikut:



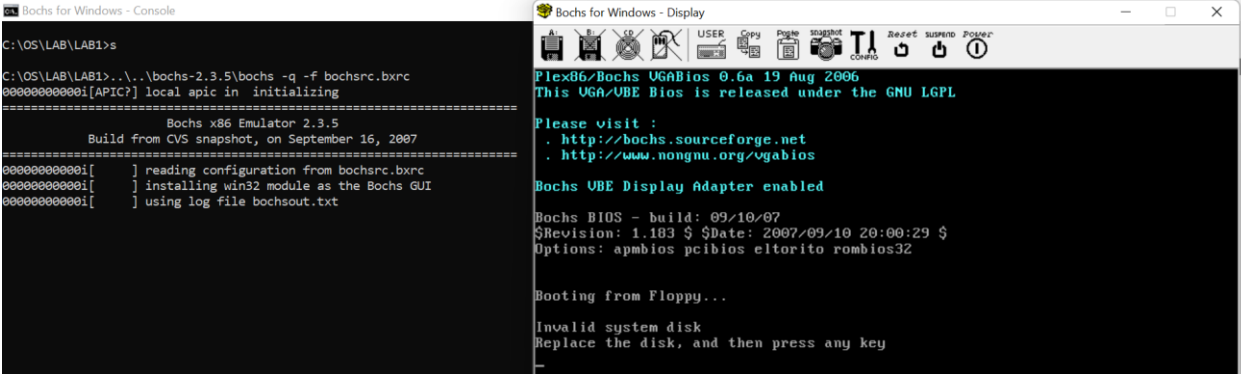
- 10. Menutup kembali PC-Simulator dengan klik pada tombol power.
- 11. mengetik ‘tdump boots.bin’ lalu tekan enter, maksud dari perintah ini adalah memindahkan datapada file ’boots.bin’ ke dalam memory kerja ‘tdump’. Seperti gambar dibawah:



- 12. Salah satu file dalam direktori kerja adalah file ‘s.bat’, berisi dua baris perintah untuk memanggil PC-Simulator ‘Bochs’. Lihat isi file ‘s.bat’ dengan perintah ‘type s.bat’ <ENTER>, hasilnya terlihat seperti pada gambar berikut:

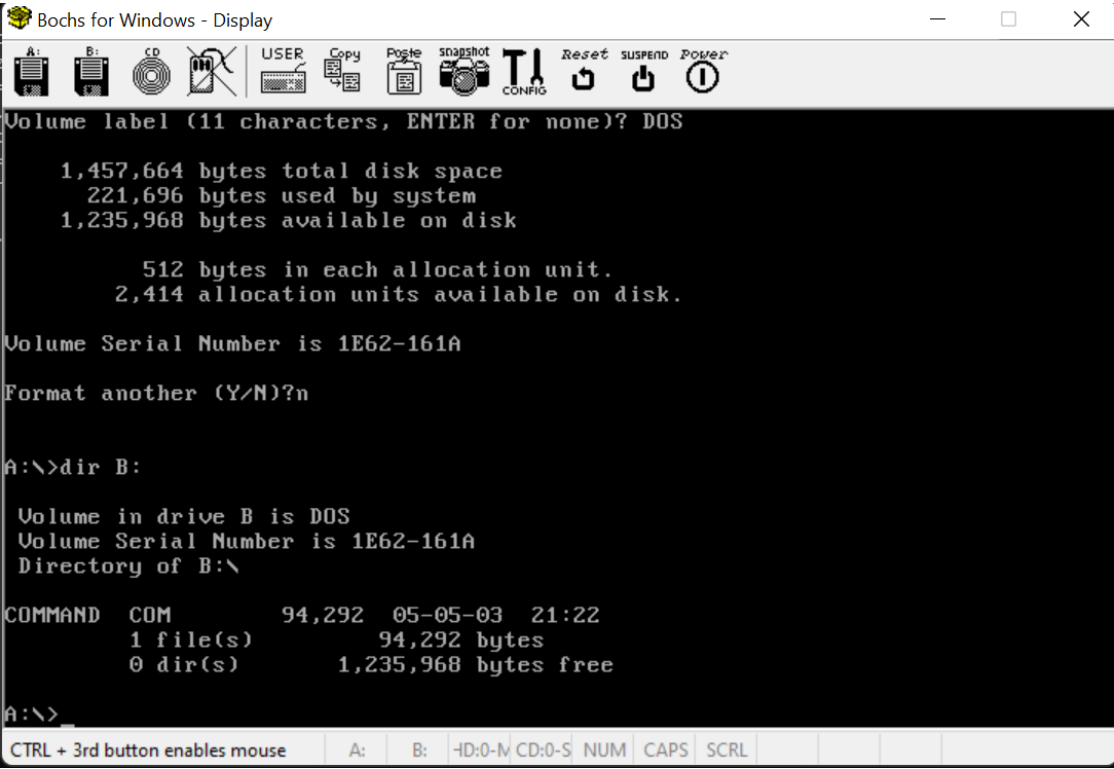
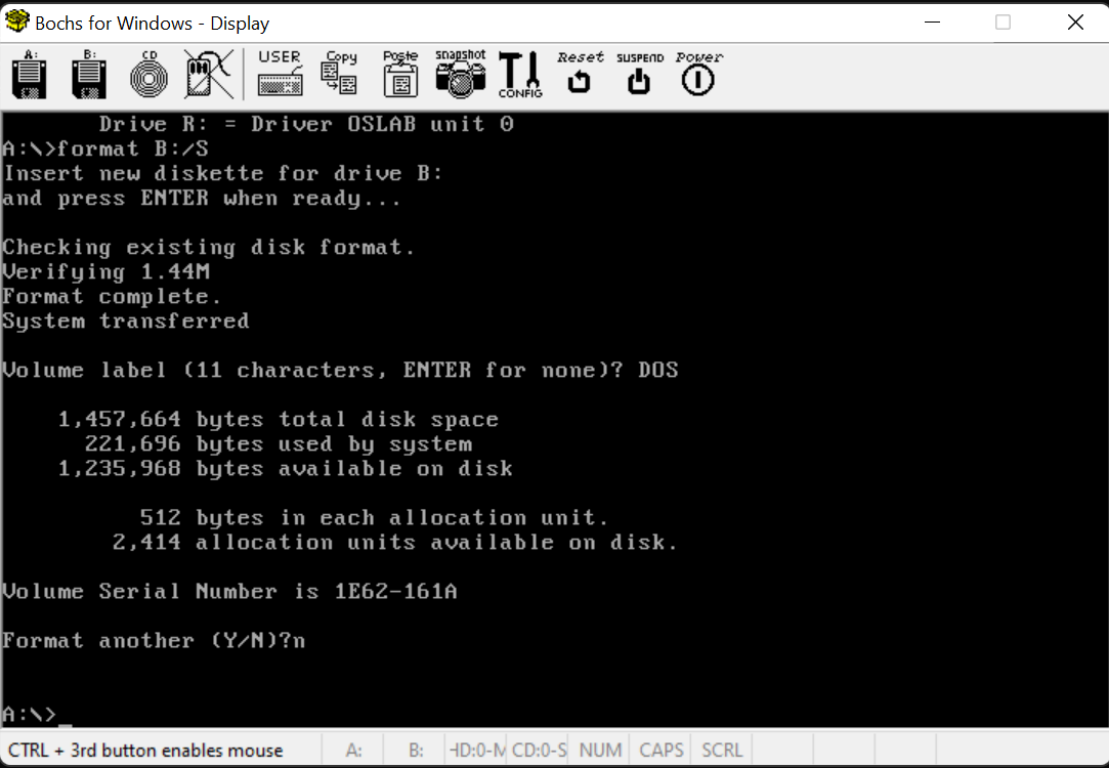


- 13. Memasukkan perintah ‘s’ <ENTER>, kemudian akan ditampilkan windows ‘Bochs for windows – display’ yang sedang melakukan proses ‘booting’ namun tidak berhasil karena tidak menemukan diskboot, seperti yang ditampilkan dibawah ini:



- 14. Untuk memformat ‘floppya.img’ dan menambahkan ‘system file’ kedalamnya, panggil ‘DosFP’ <ENTER>. Pada windows ‘Bochs’ masukan perintah ‘A:>format B:/S’ <ENTER>, selesaikan proses format dan berikan nama label disk jika perlu. Jika sudah selesai, untuk memastikan bahwa floppy pada drive ‘B:’ terisi dengan ‘system file’, periksa

dengan perintah berikut ‘A:>dir B:’ <ENTER>, jika tidak ada kesalahan pada windows ‘Bochs’ akan ditampilkan berita berikut :



15. Yang terakhir, matikan PC-Simulator (klik tombol Power Off), sekarang kita coba untuk menggunakan ‘floppya.img’ sebagai ‘boot disk’ lagi, ketik ‘S’ <ENTER>. Jika tidak ada kesalahan akan ditampilkan teks seperti berikut:

C:\OS\LAB\LAB1>s



Plex86/Bochs UGABios 0.6a 19 Aug 2006
This UGA/UBE Bios is released under the GNU LGPL

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

Bochs UBE Display Adapter enabled

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

Booting from Floppy...
Starting BootCD.....

Microsoft(R) MS-DOS 7.1
(C)Copyright Microsoft Corp 1981-1999.

A:\>_