

Nama : Nadya Ayu Widya

NIM : L200180099

## MODUL 1

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. Pada materi kali ini sobat akan menemukan 8 bit, 256 karakter ASCII, menurut **ISO 8859-1** dan **Microsoft Windows Latin-1** dengan peningkatan karakter, yang tersedia dalam program tertentu seperti Microsoft Word.

Dalam bahasa komputer 0 dan 1 tidak ada cara lain untuk mewakili huruf dan karakter yang bukan nomer. Semuanya harus menggunakan 0 dan 1. Salah satu jalan untuk berbahasa dengan komputer dengan cara menggunakan tabel ASCII. Tabel ASCII merupakan tabel atau daftar yang berisi semua huruf dalam alfabet romawi ditambah beberapa karakter tambahan. Dalam tabel ini setiap karakter akan selalu diwakili oleh sejumlah kode yang sama. Misal untuk huruf "b" (b kecil) selalu diwakili oleh urutan nomer 98, dan kalo dipresentasi menggunakan 0 dan 1 dalam bilangan biner, 98 adalah bilangan biner 110 0010.

### TABEL ASCII

Character Name	Char	Code	Decimal	Binary	Hex
Null	NUL	Ctrl @	0	00000000	00
Start of Heading	SOH	Ctrl A	1	00000001	01
Start of Text	STX	Ctrl B	2	00000010	02
End of Text	ETX	Ctrl C	3	00000011	03
End of Transmit	EOT	Ctrl D	4	00000100	04
Enquiry	ENQ	Ctrl E	5	00000101	05
Acknowledge	ACK	Ctrl F	6	00000110	06
Bell	BEL	Ctrl G	7	00000111	07
Back Space	BS	Ctrl H	8	00001000	08
Horizontal Tab	TAB	Ctrl I	9	00001001	09
Line Feed	LF	Ctrl J	10	00001010	0A
Vertical Tab	VT	Ctrl K	11	00001011	0B
Form Feed	FF	Ctrl L	12	00001100	0C
Carriage Return	CR	Ctrl M	13	00001101	0D
Shift Out	SO	Ctrl N	14	00001110	0E

<b>Shift In</b>	SI	Ctrl O	15	00001111	0F
<b>Data Line Escape</b>	DLE	Ctrl P	16	00010000	10
<b>Device Control 1</b>	DC1	Ctrl Q	17	00010001	11
<b>Device Control 2</b>	DC2	Ctrl R	18	00010010	12
<b>Device Control 3</b>	DC3	Ctrl S	19	00010011	13
<b>Device Control 4</b>	DC4	Ctrl T	20	00010100	14
<b>Negative Acknowledge</b>	NAK	Ctrl U	21	00010101	15
<b>Synchronous Idle</b>	SYN	Ctrl V	22	00010110	16
<b>End of Transmit Block</b>	ETB	Ctrl W	23	00010111	17
<b>Cancel</b>	CAN	Ctrl X	24	00011000	18
<b>End of Medium</b>	EM	Ctrl Y	25	00011001	19
<b>Substitute</b>	SUB	Ctrl Z	26	00011010	1A
<b>Escape</b>	ESC	Ctrl [	27	00011011	1B
<b>File Separator</b>	FS	Ctrl \	28	00011100	1C
<b>Group Separator</b>	GS	Ctrl ]	29	00011101	1D
<b>Record Separator</b>	RS	Ctrl ^	30	00011110	1E
<b>Unit Separator</b>	US	Ctrl _	31	00011111	1F
<b>Space</b>			32	00100000	20
<b>Exclamation Point</b>	!	Shift 1	33	00100001	21
<b>Double Quote</b>	"	Shift ‘	34	00100010	22
<b>Pound/Number Sign</b>	#	Shift 3	35	00100011	23
<b>Dollar Sign</b>	\$	Shift 4	36	00100100	24
<b>Percent Sign</b>	%	Shift 5	37	00100101	25
<b>Ampersand</b>	&	Shift 7	38	00100110	26
<b>Single Quote</b>	‘	‘	39	00100111	27
<b>Left Parenthesis</b>	(	Shift 9	40	00101000	28
<b>Right Parenthesis</b>	)	Shift 0	41	00101001	29
<b>Asterisk</b>	*	Shift 8	42	00101010	2A

<b>Plus Sign</b>	+	Shift =	43	00101011	2B
<b>Comma</b>	,	,	44	00101100	2C
<b>Hyphen / Minus Sign</b>	-	-	45	00101101	2D
<b>Period</b>	.	.	46	00101110	2E
<b>Forward Slash</b>	/	/	47	00101111	2F
<b>Zero Digit</b>	0	0	48	00110000	30
<b>One Digit</b>	1	1	49	00110001	31
<b>Two Digit</b>	2	2	50	00110010	32
<b>Three Digit</b>	3	3	51	00110011	33
<b>Four Digit</b>	4	4	52	00110100	34
<b>Five Digit</b>	5	5	53	00110101	35
<b>Six Digit</b>	6	6	54	00110110	36
<b>Seven Digit</b>	7	7	55	00110111	37
<b>Eight Digit</b>	8	8	56	00111000	38
<b>Nine Digit</b>	9	9	57	00111001	39
<b>Colon</b>	:	Shift ;	58	00111010	3A
<b>Semicolon</b>	;	;	59	00111011	3B
<b>Less-Than Sign</b>	<	Shift ,	60	00111100	3C
<b>Equals Sign</b>	=	=	61	00111101	3D
<b>Greater-Than Sign</b>	>	Shift .	62	00111110	3E
<b>Question Mark</b>	?	Shift /	63	00111111	3F
<b>At Sign</b>	@	Shift 2	64	01000000	40
<b>Capital A</b>	A	Shift A	65	01000001	41
<b>Capital B</b>	B	Shift B	66	01000010	42
<b>Capital C</b>	C	Shift C	67	01000011	43
<b>Capital D</b>	D	Shift D	68	01000100	44
<b>Capital E</b>	E	Shift E	69	01000101	45
<b>Capital F</b>	F	Shift F	70	01000110	46

<b>Capital G</b>	G	Shift G	71	01000111	47
<b>Capital H</b>	H	Shift H	72	01001000	48
<b>Capital I</b>	I	Shift I	73	01001001	49
<b>Capital J</b>	J	Shift J	74	01001010	4A
<b>Capital K</b>	K	Shift K	75	01001011	4B
<b>Capital L</b>	L	Shift L	76	01001100	4C
<b>Capital M</b>	M	Shift M	77	01001101	4D
<b>Capital N</b>	N	Shift N	78	01001110	4E
<b>Capital O</b>	O	Shift O	79	01001111	4F
<b>Capital P</b>	P	Shift P	80	01010000	50
<b>Capital Q</b>	Q	Shift Q	81	01010001	51
<b>Capital R</b>	R	Shift R	82	01010010	52
<b>Capital S</b>	S	Shift S	83	01010011	53
<b>Capital T</b>	T	Shift T	84	01010100	54
<b>Capital U</b>	U	Shift U	85	01010101	55
<b>Capital V</b>	V	Shift V	86	01010110	56
<b>Capital W</b>	W	Shift W	87	01010111	57
<b>Capital X</b>	X	Shift X	88	01011000	58
<b>Capital Y</b>	Y	Shift Y	89	01011001	59
<b>Capital Z</b>	Z	Shift Z	90	01011010	5A
<b>Left Bracket</b>	[	[	91	01011011	5B
<b>Backward Slash</b>	\	\	92	01011100	5C
<b>Right Bracket</b>	]	]	93	01011101	5D

<b>Caret</b>	^	Shift 6	94	01011110	5E
<b>Underscore</b>	_	Shift -	95	01011111	5F
<b>Back Quote</b>	`	`	96	01100000	60
<b>Lower-case A</b>	a	A	97	01100001	61
<b>Lower-case B</b>	b	B	98	01100010	62
<b>Lower-case C</b>	c	C	99	01100011	63
<b>Lower-case D</b>	d	D	100	01100100	64
<b>Lower-case E</b>	e	E	101	01100101	65
<b>Lower-case F</b>	f	F	102	01100110	66
<b>Lower-case G</b>	g	G	103	01100111	67
<b>Lower-case H</b>	h	H	104	01101000	68
<b>Lower-case I</b>	i	I	105	01101001	69
<b>Lower-case J</b>	j	J	106	01101010	6A
<b>Lower-case K</b>	k	K	107	01101011	6B
<b>Lower-case L</b>	l	L	108	01101100	6C
<b>Lower-case M</b>	m	M	109	01101101	6D
<b>Lower-case N</b>	n	N	110	01101110	6E
<b>Lower-case O</b>	o	O	111	01101111	6F
<b>Lower-case P</b>	p	P	112	01110000	70
<b>Lower-case Q</b>	q	Q	113	01110001	71
<b>Lower-case R</b>	r	R	114	01110010	72
<b>Lower-case S</b>	s	S	115	01110011	73
<b>Lower-case T</b>	t	T	116	01110100	74
<b>Lower-case U</b>	u	U	117	01110101	75
<b>Lower-case V</b>	v	V	118	01110110	76
<b>Lower-case W</b>	w	W	119	01110111	77
<b>Lower-case X</b>	x	X	120	01111000	78
<b>Lower-case Y</b>	y	Y	121	01111001	79
<b>Lower-case Z</b>	z	Z	122	01111010	7A
<b>Left Brace</b>	{	Shift [	123	01111011	7B

<b>Vertical Bar</b>		Shift \	124	01111100	7C
<b>Right Brace</b>	}	Shift ]	125	01111101	7D
<b>Tilde</b>	~	Shift `	126	01111110	7E
<b>Delta</b>	□		127	01111111	7F

2. Perintah bahasa Assembly untuk mesin intel X86 lengkap.

#### Daftar Assembly Directive

Assembly Directive	Keterangan
EQU	Pendefinisian konstanta
DB	Pendefinisian data dengan ukuran satuan 1 byte
DW	Pendefinisian data dengan ukuran satuan 1 word
DBIT	Pendefinisian data dengan ukuran satuan 1 bit
DS	Pemesanan tempat penyimpanan data di RAM
ORG	Inisialisasi alamat mulai program
END	Penanda akhir program
CSEG	Penanda penempatan di code segment
XSEG	Penanda penempatan di external data segment
DSEG	Penanda penempatan di internal direct data segment
ISEG	Penanda penempatan di internal indirect data segment
BSEG	Penanda penempatan di bit data segment
CODE	Penanda mulai pendefinisian program
XDATA	Pendefinisian external data
DATA	Pendefinisian internal direct data
IDATA	Pendefinisian internal indirect data
BIT	Pendefinisian data bit
#INCLUDE	Mengikutsertakan file program lain

#### Daftar Instruksi

Instruksi	Keterangan Singkatan
ACALL	Absolute Call
ADD	Add
ADDC	Add with Carry
AJMP	Absolute Jump

ANL	AND Logic
CJNE	Compare and Jump if Not Equal
CLR	Clear
CPL	Complement
DA	Decimal Adjust
DEC	Decrement
DIV	Divide
DJNZ	Decrement and Jump if Not Zero
INC	Increment
JB	Jump if Bit Set
JBC	Jump if Bit Set and Clear Bit
JC	Jump if Carry Set
JMP	Jump to Address
JNB	Jump if Not Bit Set
JNC	Jump if Carry Not Set
JNZ	Jump if Accumulator Not Zero
JZ	Jump if Accumulator Zero
LCALL	Long Call
LJMP	Long Jump
MOV	Move from Memory
MOVC	Move from Code Memory
MOVB	Move from Extended Memory
MUL	Multiply
NOP	No Operation
ORL	OR Logic
POP	Pop Value From Stack
PUSH	Push Value Onto Stack
RET	Return From Subroutine
RETI	Return From Interrupt
RL	Rotate Left
RLC	Rotate Left through Carry
RR	Rotate Right
RRC	Rotate Right through Carry
SETB	Set Bit
SJMP	Short Jump

SUBB	Subtract With Borrow
SWAP	Swap Nibbles
XCH	Exchange Bytes
XCHD	Exchange Digits
XRL	Exclusive OR Logic