

MODUL 1

1. Kode Standar Amerika untuk Pertukaran Informasi atau American Standard Code for Information Interchange (ASCII) merupakan suatu standar internasional dalam kode huruf dan simbol seperti Hex dan Unicode tetapi ASCII lebih bersifat universal, contohnya 124 adalah untuk karakter "|". Ia selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. Kode ASCII sebenarnya memiliki komposisi bilangan biner sebanyak 7 bit. Namun, ASCII disimpan sebagai sandi 8 bit dengan menambahkan satu angka 0 sebagai bit significant paling tinggi. Bit tambahan ini sering digunakan untuk uji paritas. Karakter control pada ASCII dibedakan menjadi 5 kelompok sesuai dengan penggunaan yaitu berturut-turut meliputi logical communication, Device control, Information separator, Code extention, dan physical communication. Code ASCII ini banyak dijumpai pada papan ketik (keyboard) computer atau instrument-instrument digital.

TABEL ASCII :

Nilai ANSI ASCII	Nilai Unicode (Heksa)	Binn	Karakt
0	0	00000000	N
1	0	00000001	SO
2	0	00000010	S
3	0	00000011	E
4	0	00000100	E
5	0	00000101	EN
6	0	00000110	A
7	0	00000111	B
8	0	00001000	B
9	0	00001001	H
1	0	00001010	L
1	0	00001011	V
1	0	00001100	F
1	0	00001101	C
1	0	00001110	S
1	0	00001111	S
1	1	00010000	D
1	1	00010001	D
1	1	00010010	D
1	1	00010011	D
2	1	00010100	D
2	1	00010101	NA
2	1	00010110	SY
2	1	00010111	E
2	1	00011000	CA
2	1	00011001	E
2	1	00011010	S
2	1	00011011	E

2	1	00011100	F
2	1	00011101	G

3	1	00011110	R
3	1	00011111	U
3	2	00100000	spa
3	2	00100001	!
3	2	00100010	"
3	2	00100011	#
3	2	00100100	\$
3	2	00100101	%
3	2	00100110	&
3	2	00100111	'
4	2	00101000	(
4	2	00101001	)
4	2	00101010	*
4	2	00101011	+
4	2	00101100	,
4	2	00101101	-
4	2	00101110	.
4	2	00101111	/
4	3	00110000	0
4	3	00110001	1
5	3	00110010	2
5	3	00110011	3
5	3	00110100	4
5	3	00110101	5
5	3	00110110	6
5	3	00110111	7
5	3	00111000	8
5	3	00111001	9
5	3	00111010	:
5	3	00111011	;
6	3	00111100	<
6	3	00111101	=
6	3	00111110	>
6	3	00111111	?
6	4	01000000	@
6	4	01000001	A
6	4	01000010	B
6	4	01000011	C
6	4	01000100	D
6	4	01000101	E
7	4	01000110	F
7	4	01000111	G
7	4	01001000	H
7	4	01001001	I
7	4	01001010	J

7	4	01001011	K
7	4	01001100	L
7	4	01001101	M
7	4	01001110	N
7	4	01001111	O
8	5	01010000	P
8	5	01010001	Q
8	5	01010010	R
8	5	01010011	S
8	5	01010100	T
8	5	01010101	U
8	5	01010110	V
8	5	01010111	W
8	5	01011000	X
8	5	01011001	Y
9	5	01011010	Z
9	5	01011011	[
9	5	01011100	\
9	5	01011101	]
9	5	01011110	^
9	5	01011111	_
9	6	01100000	`
9	6	01100001	a
9	6	01100010	b
9	6	01100011	c
1	6	01100100	d
1	6	01100101	e
1	6	01100110	f
1	6	01100111	g
1	6	01101000	h
1	6	01101001	i
1	6	01101010	j
1	6	01101011	k
1	6	01101100	l
1	6	01101101	m
1	6	01101110	n
1	6	01101111	o
1	7	01110000	p
1	7	01110001	q
1	7	01110010	r
1	7	01110011	s
1	7	01110100	t
1	7	01110101	u
1	7	01110110	v
1	7	01110111	w

1	7	01111000	x
1	7	01111001	y
1	7	01111010	z
1	7	01111011	{
1	7	01111100	
1	7	01111101	}
1	7	01111110	~
1	7	01111111	D

2. Daftar Perintah Bahasa Assembly:  
Perintah bahasa assembly lengkap untuk mesin intel keluarga dan 86. Digunakan sebagai pedoman untuk memahami program boot.asm dan kernel.asm.  
Assembly Directive

- EQU => Detinisian konstanta
- DB => Definisian data dengan satuan 1 byte
- DW => Definisian data dengan satuan 1 word
- DBIT => Pendefinisian data dengan satuan 1 bit
- DS => Pemesanan tempat penyimpanan data di PAM
- ORG => Insiluasi alamat mulai program
- EHD => Penanda akhir program
- CSGC => Penempatan penanda dicode segment
- XSEG => Penanda penempatan di eksternal data
- DSEG => Penanda penempatan internal direct data
- 1SEG => Penanda penempatan di internal indirect data
- 13SEG => Penanda penempatan di bit data
- CODE => Penanda mulai pendefinisian program
- XDATA => Pendefinisian external data
- 1DATA => Pendefinisian indirect data
- BIT => Pendefinisian data bit
- INCLUDE => Mengikut sertakan file program lain
- DATA => Pendefinisian internal direct data

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