Nama : Alfianto Andy Pamungkas

NIM : L200180194

MODUL 1

1. ASCII merupakan kepanjangan dari (American Standard Code for Information Interchange), dan pengertian dari ASCII sendiri adalah 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.

   Tabel ASCII

Dec	Hex	Binary	HTML	Char	Description
0	00	00000000	&#0;	NUL	Null
1	01	00000001	&#1;	SOH	Start of Header
2	02	00000010	&#2;	STX	Start of Text
3	03	00000011	&#3;	ETX	End of Text
4	04	00000100	&#4;	EOT	End of Transmission
5	05	00000101	&#5;	ENQ	Enquiry
6	06	00000110	&#6;	ACK	Acknowledge
7	07	00000111	&#7;	BEL	Bell
8	08	00001000	&#8;	BS	Backspace
9	09	00001001		HT	Horizontal Tab
10	0A	00001010
LF	Line Feed
11	0B	00001011	&#11;	VT	Vertical Tab
12	0C	00001100		FF	Form Feed
13	0D	00001101
CR	Carriage Return
14	0E	00001110	&#14;	SO	Shift Out
15	0F	00001111	&#15;	SI	Shift In
16	10	00010000	&#16;	DLE	Data Link Escape
17	11	00010001	&#17;	DC1	Device Control 1
18	12	00010010	&#18;	DC2	Device Control 2
19	13	00010011	&#19;	DC3	Device Control 3
20	14	00010100	&#20;	DC4	Device Control 4
21	15	00010101	&#21;	NAK	Negative Acknowledge
22	16	00010110	&#22;	SYN	Synchronize

23	17	00010111	&#23;	ETB	End of Transmission Block
24	18	00011000	&#24;	CAN	Cancel
25	19	00011001	&#25;	EM	End of Medium
26	1A	00011010	&#26;	SUB	Substitute
27	1B	00011011	&#27;	ESC	Escape
28	1C	00011100	&#28;	FS	File Separator
29	1D	00011101	&#29;	GS	Group Separator
30	1E	00011110	&#30;	RS	Record Separator
31	1F	00011111	&#31;	US	Unit Separator
32	20	00100000	&#32;	space	Space
33	21	00100001	&#33;	!	Exclamation mark
34	22	00100010	&#34;	"	Double quote
35	23	00100011	&#35;	#	Number
36	24	00100100	&#36;	$	Dollar sign
37	25	00100101	&#37;	%	Percent
38	26	00100110	&#38;	&	Ampersand
39	27	00100111	&#39;	'	Single quote
40	28	00101000	&#40;	(	Left parenthesis
41	29	00101001	&#41;	)	Right parenthesis
42	2A	00101010	&#42;	*	Asterisk
43	2B	00101011	&#43;	+	Plus
44	2C	00101100	&#44;	,	Comma
45	2D	00101101	&#45;	-	Minus
46	2E	00101110	&#46;	.	Period
47	2F	00101111	&#47;	/	Slash
48	30	00110000	&#48;	0	Zero
49	31	00110001	&#49;	1	One
50	32	00110010	&#50;	2	Two
51	33	00110011	&#51;	3	Three
52	34	00110100	&#52;	4	Four
53	35	00110101	&#53;	5	Five
54	36	00110110	&#54;	6	Six
55	37	00110111	&#55;	7	Seven
56	38	00111000	&#56;	8	Eight
57	39	00111001	&#57;	9	Nine

58	3A	00111010	&#58;	:	Colon
59	3B	00111011	&#59;	;	Semicolon
60	3C	00111100	&#60;	<	Less than
61	3D	00111101	&#61;	=	Equality sign
62	3E	00111110	&#62;	>	Greater than
63	3F	00111111	&#63;	?	Question mark
64	40	01000000	&#64;	@	At sign
65	41	01000001	&#65;	A	Capital A
66	42	01000010	&#66;	B	Capital B
67	43	01000011	&#67;	C	Capital C
68	44	01000100	&#68;	D	Capital D
69	45	01000101	&#69;	E	Capital E
70	46	01000110	&#70;	F	Capital F
71	47	01000111	&#71;	G	Capital G
72	48	01001000	&#72;	H	Capital H
73	49	01001001	&#73;	I	Capital I
74	4A	01001010	&#74;	J	Capital J
75	4B	01001011	&#75;	K	Capital K
76	4C	01001100	&#76;	L	Capital L
77	4D	01001101	&#77;	M	Capital M
78	4E	01001110	&#78;	N	Capital N
79	4F	01001111	&#79;	O	Capital O
80	50	01010000	&#80;	P	Capital P
81	51	01010001	&#81;	Q	Capital Q
82	52	01010010	&#82;	R	Capital R
83	53	01010011	&#83;	S	Capital S
84	54	01010100	&#84;	T	Capital T
85	55	01010101	&#85;	U	Capital U
86	56	01010110	&#86;	V	Capital V
87	57	01010111	&#87;	W	Capital W
88	58	01011000	&#88;	X	Capital X
89	59	01011001	&#89;	Y	Capital Y
90	5A	01011010	&#90;	Z	Capital Z
91	5B	01011011	&#91;	[	Left square bracket
92	5C	01011100	&#92;	\	Backslash

93	5D	01011101	&#93;	]	Right square bracket
94	5E	01011110	&#94;	^	Caret / circumflex
95	5F	01011111	&#95;	_	Underscore
96	60	01100000	&#96;	`	Grave / accent
97	61	01100001	&#97;	a	Small a
98	62	01100010	&#98;	b	Small b
99	63	01100011	&#99;	c	Small c
100	64	01100100	&#100;	d	Small d
101	65	01100101	&#101;	e	Small e
102	66	01100110	&#102;	f	Small f
103	67	01100111	&#103;	g	Small g
104	68	01101000	&#104;	h	Small h
105	69	01101001	&#105;	i	Small i
106	6A	01101010	&#106;	j	Small j
107	6B	01101011	&#107;	k	Small k
108	6C	01101100	&#108;	l	Small l
109	6D	01101101	&#109;	m	Small m
110	6E	01101110	&#110;	n	Small n
111	6F	01101111	&#111;	o	Small o
112	70	01110000	&#112;	p	Small p
113	71	01110001	&#113;	q	Small q
114	72	01110010	&#114;	r	Small r
115	73	01110011	&#115;	s	Small s
116	74	01110100	&#116;	t	Small t
117	75	01110101	&#117;	u	Small u
118	76	01110110	&#118;	v	Small v
119	77	01110111	&#119;	w	Small w
120	78	01111000	&#120;	x	Small x
121	79	01111001	&#121;	y	Small y
122	7A	01111010	&#122;	z	Small z
123	7B	01111011	&#123;	{	Left curly bracket
124	7C	01111100	&#124;	|	Vertical bar
125	7D	01111101	&#125;	}	Right curly bracket
126	7E	01111110	&#126;	~	Tilde
127	7F	01111111	&#127;	DEL	Delete

2. *Daftar instruksi bahasa Assembly pada x86*

Dalam program bahasa assembly terdapat 2 jenis yang kita tulis dalam program:

- ***Assembly Directive*** *(yaitu merupakan kode yang menjadi arahan bagi assembler/compiler untuk menata program).*
- ***Instruksi*** *(yaitu kode yang harus dieksekusi oleh CPU mikrokontroler dengan melakukan operasi tertentu sesuai dengan daftar yang sudah tertanam dalam CPU) ,*

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*
MOVX	*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*

untuk yang lebih jelas dan detil:

a. MOV

Perintah MOV adalah perintah untuk mengisi, memindahkan,memperbaruhi isi suatu register, variable ataupun lokasi memory, Adapun tata penulisan perintah MOV adalah :

MOV [operand A], [Operand B]

Contoh :

MOV AH,02

Operand A adalah Register AH

Operand B adalah bilangan 02

Hal yang dilakukan oleh komputer untuk perintah diatas adalahmemasukan 02 ke register AH.

b. INT (Interrupt)

Bila anda pernah belajar BASIC, maka pasti anda tidak asing lagi dengan perintah GOSUB. Perintah INT juga mempunyai cara kerja yang sama dengan GOSUB, hanya saja subroutine yang dipanggil telah disediakan oleh memory komputer yang terdiri 2 jenis yaitu :

- Bios Interrupt (interput yang disediakan oleh BIOS (INT 0 – INT 1F))

- Dos Interrupt (Interrupt yang disediakan oleh DOS (INT 1F – keatas))

c. Push

Adalah perintah untuk memasukan isi register pada stack, dengan tata penulisannya:POP [operand 16 bit]

d. Pop

perintah yang berguna untuk mengeluarkan isi dari register/variable dari stack,dengan tata penulisannya adalah : POP [operand 16 bit]

e. RIP (Register IP)

Perintah ini digunakan untuk memberitahu komputer untuk memulai memproses program dari titik tertentu.

f. A (Assembler)

Perintah Assembler berguna untuk tempat menulis program Assembler.
-A100
0FD8:100

g. RCX (Register CX)

Perintah ini digunakan untuk mengetahui dan memperbaruhi isi register CX yang merupakantempat penampungan panjang program yang sedang aktif