NAMA : CECYLIA IVANKA HERMANITA
NIM : L200180210

TUGAS MODUL

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 99 adalah untuk karakter "c". Ia selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. Sedangkan fungsi dari kode ASCII ialah digunakan untuk mewakili karakter-karakter angka maupun huruf didalam komputer, sebagai contoh dapat kita lihat pada karakter 1, 2, 3, A, B, C, dan sebagainya.

TABEL KODE ASCII STANDAR

DEC	BINER	HEX	SYMBOL	DEC	BINER	HEX	SYMBOL	DEC	BINER	HEX	SYMBOL	DEC	BINER	HEX	SYMBOL
0	00000000	00	NUL	32	00100000	20		64	01000000	40	@	96	01100000	60	`
1	00000001	01	SOH	33	00100001	21	!	65	01000001	41	A	97	01100001	61	a
2	00000010	02	STX	34	00100010	22	"	66	01000010	42	B	98	01100010	62	b
3	00000011	03	ETX	35	00100011	23	#	67	01000011	43	C	99	01100011	63	c
4	00000100	04	EOT	36	00100100	24	$	68	01000100	44	D	100	01100100	64	d
5	00000101	05	ENQ	37	00100101	25	%	69	01000101	45	E	101	01100101	65	e
6	00000110	06	ACK	38	00100110	26	&	70	01000110	46	F	102	01100110	66	f
7	00000111	07	BEL	39	00100111	27	'	71	01000111	47	G	103	01100111	67	g
8	00001000	08	BS	40	00101000	28	(	72	01001000	48	H	104	01101000	68	h
9	00001001	09	HT	41	00101001	29	)	73	01001001	49	I	105	01101001	69	i
10	00001010	0A	LF	42	00101010	2A	*	74	01001010	4A	J	106	01101010	6A	j
11	00001011	0B	VT	43	00101011	2B	+	75	01001011	4B	K	107	01101011	6B	k
12	00001100	0C	FF	44	00101100	2C	,	76	01001100	4C	L	108	01101100	6C	l
13	00001101	0D	CR	45	00101101	2D	-	77	01001101	4D	M	109	01101101	6D	m
14	00001110	0E	SO	46	00101110	2E	.	78	01001110	4E	N	110	01101110	6E	n
15	00001111	0F	SI	47	00101111	2F	/	79	01001111	4F	O	111	01101111	6F	o
16	00010000	10	DLE	48	00110000	30	0	80	01010000	50	P	112	01110000	70	p
17	00010001	11	DC1	49	00110001	31	1	81	01010001	51	Q	113	01110001	71	q
18	00010010	12	DC2	50	00110010	32	2	82	01010010	52	R	114	01110010	72	r
19	00010011	13	DC3	51	00110011	33	3	83	01010011	53	S	115	01110011	73	s
20	00010100	14	DC4	52	00110100	34	4	84	01010100	54	T	116	01110100	74	t
21	00010101	15	NAK	53	00110101	35	5	85	01010101	55	U	117	01110101	75	u
22	00010110	16	SYN	54	00110110	36	6	86	01010110	56	V	118	01110110	76	v
23	00010111	17	ETB	55	00110111	37	7	87	01010111	57	W	119	01110111	77	w
24	00011000	18	CAN	56	00111000	38	8	88	01011000	58	X	120	01111000	78	x
25	00011001	19	EM	57	00111001	39	9	89	01011001	59	Y	121	01111001	79	y
26	00011010	1A	SUB	58	00111010	3A	:	90	01011010	5A	Z	122	01111010	7A	z
27	00011011	1B	ESC	59	00111011	3B	;	91	01011011	5B	[	123	01111011	7B	{
28	00011100	1C	FS	60	00111100	3C	<	92	01011100	5C	\	124	01111100	7C	|
29	00011101	1D	GS	61	00111101	3D	=	93	01011101	5D	]	125	01111101	7D	}
30	00011110	1E	RS	62	00111110	3E	>	94	01011110	5E	^	126	01111110	7E	~
31	00011111	1F	US	63	00111111	3F	?	95	01011111	5F	_	127	01111111	7F	

2. Daftar Perintah Bahasa Assembly
    1. ACALL (Absolute Call)
    2. ADD (Add Immediate Data)
    3. ADDC
    4. AJMP (Absolute Jump)
    5. ANL (logical AND memori ke akumulator)
    6. CJNE (Compare Indirect Address to Immediate Data)
    7. CLR (Clear Accumulator)
    8. CPL (Complement Accumulator)
    9. DA (Decimal Adjust Accumulator)
    10. DEC (Decrement Indirect Address)
    11. DIV (Divide Accumulator by B)
    12. DJNZ (Decrement Register and Jump Id Not Zero)
    13. INC (Increment Indirect Address)
    14. JB (Jump if Bit is Set)
    15. JBC (Jump if Bit Set and Clear Bit)
    16. JC (Jump if Carry is Set)
    17. JMP (Jump to sum of Accumulator and Data Pointer)
    18. JNB (Jump if Bit is Not Set)
    19. JNC (Jump if Carry Not Set)
    20. JNZ (Jump if Accumulator Not Zero)
    21. JZ (Jump if Accumulator is Zero)
    22. LCALL (Long Call)
    23. LJMP (Long Jump)
    24. MOV (Move from Memory)
    25. MOVC (Move from Codec Memory)
    26. MOVX (Move Accumulator to External Memory Addressed by Data Pointer)
    27. MUL (Multiply)
    28. NOP (No Operation)
    29. ORL (Logical OR Immediate Data to Accumulator)
    30. POP (Pop Stack to Memory)
    31. PUSH (Push Memory onto Stack)

32. RET (Return from subroutine)
33. RETI (Return from Interrupt)
34. RL (Rotate Accumulator Left)
35. RLC (Rotate Left through Carry)
36. RR (Rotate Right)
37. RRC (Rotate Right through Carry)
38. SETB (set Carry flag)
39. SJMP (Short Jump)
40. SUBB (Subtract With Borrow)
41. SWAP (Swap Nibbles)
42. XCH (Exchange Bytes)
43. XCHD (Exchange Digits)
44. XRL (Exclusive OR Logic)