Nama : Riska Meilina Puspa

NIM : L200180192

MODUL 1

ASCII merupakan kepanjangan dari (***American Standard Code for Information Interchange***), dan **ASCII** sendiri adalah suatu standar internasional dalam kode huruf dan simbol seperti Hex dan Unicode tetapi ASCII lebih bersifat universal. 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 menambakan satu angka 0 sebagai bit significant paling tinggi. Bit tambahan ini sering digunakan untuk uji paritas.

Decimal	Hexadesimal	Binary	Character	Description
32	20	00100000	**Space**	**space**
33	21	00100001	**!**	exclamation mark
34	22	00100010	"	double quote
35	23	00100011	#	number
36	24	00100100	**$**	dollar
37	25	00100101	**%**	percent
38	26	00100110	**&**	ampersand
39	27	00100111	'	single quote
40	28	00101000	**(**	left parenthesis
41	29	00101001	**)**	right parenthesis
42	2A	00101010	*****	asterisk

Decimal	Hexadesimal	Binary	Character	Description
43	2B	00101011	**+**	plus
44	2C	00101100	**,**	comma
45	2D	00101101	**-**	minus
46	2E	00101110	**.**	period
47	2F	00101111	**/**	slash
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
61	3D	00111101	**=**	equality sign
62	3E	00111110	>	greater than

Decimal	Hexadesimal	Binary	Character	Description
63	3F	00111111	**?**	question mark
64	40	01000000	**@**	at sign
65	41	01000001	**A**	
66	42	01000010	**B**	
67	43	01000011	**C**	
68	44	01000100	**D**	
69	45	01000101	**E**	
70	46	01000110	**F**	
71	47	01000111	**G**	
72	48	01001000	**H**	
73	49	01001001	**I**	
74	4A	01001010	**J**	
75	4B	01001011	**K**	
76	4C	01001100	**L**	
77	4D	01001101	**M**	
78	4E	01001110	**N**	
79	4F	01001111	**O**	
80	50	01010000	**P**	
81	51	01010001	**Q**	
82	52	01010010	**R**	

Decimal	Hexadesimal	Binary	Character	Description
83	53	01010011	**S**	
84	54	01010100	**T**	
85	55	01010101	**U**	
86	56	01010110	**V**	
87	57	01010111	**W**	
88	58	01011000	**X**	
89	59	01011001	**Y**	
90	5A	01011010	**Z**	
91	5B	01011011	**[**	left square bracket
92	5C	01011100	\	backslash
93	5D	01011101	**]**	right square bracket
94	5E	01011110	^	caret / circumflex
95	5F	01011111	**_**	underscore
96	60	01100000	**`**	grave / accent
97	61	01100001	**a**	
98	62	01100010	**b**	
99	63	01100011	**c**	
100	64	01100100	**d**	
101	65	01100101	**e**	
102	66	01100110	**f**	

Decimal	Hexadesimal	Binary	Character	Description
103	67	01100111	**g**	
104	68	01101000	**h**	
105	69	01101001	**i**	
106	6A	01101010	**j**	
107	6B	01101011	**k**	
108	6C	01101100	**l**	
109	6D	01101101	**m**	
110	6E	01101110	**n**	
111	6F	01101111	**o**	
112	70	01110000	**p**	
113	71	01110001	**q**	
114	72	01110010	**r**	
115	73	01110011	**s**	
116	74	01110100	**t**	
117	75	01110101	**u**	
118	76	01110110	**v**	
119	77	01110111	**w**	
120	78	01111000	**x**	
121	79	01111001	**y**	
122	7A	01111010	**z**	

Decimal	Hexadesimal	Binary	Character	Description
123	7B	01111011	**{**	left curly bracket
124	7C	01111100	|	vertical bar
125	7D	01111101	**}**	right curly bracket
126	7E	01111110	~	tilde
127	7F	01111111	DEL	delete

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 merupakan tempat penampungan panjang program yang sedang aktif