Nama : Damping Rizki Wisnu Adji
NIM : L200180213
Kelas E

NO. 1

Karakter	Nilai Unicode (heksadesimal)	Nilai ANSI ASCII (desimal)	Keterangan
NUL	0000	0	Null (tidak tampak)
SOH	0001	1	Start of heading (tidak tampak)
STX	0002	2	Start of text (tidak tampak)
ETX	0003	3	End of text (tidak tampak)
EOT	0004	4	End of transmission (tidak tampak)
ENQ	0005	5	Enquiry (tidak tampak)
ACK	0006	6	Acknowledge (tidak tampak)
BEL	0007	7	Bell (tidak tampak)
BS	0008	8	Menghapus satu karakter di belakang kursor (Backspace)
HT	0009	9	Horizontal tabulation
LF	000A	10	Pergantian baris (Line feed)
VT	000B	11	Tabulasi vertikal
FF	000C	12	Pergantian baris (Form feed)
CR	000D	13	Pergantian baris (carriage return)
SO	000E	14	Shift out (tidak tampak)
SI	000F	15	Shift in (tidak tampak)
DLE	0010	16	Data link escape (tidak tampak)
DC1	0011	17	Device control 1 (tidak tampak)
DC2	0012	18	Device control 2 (tidak tampak)
DC3	0013	19	Device control 3 (tidak tampak)
DC4	0014	20	Device control 4 (tidak tampak)
NAK	0015	21	Negative acknowledge (tidak tampak)
SYN	0016	22	Synchronous idle (tidak tampak)
ETB	0017	23	End of transmission block (tidak tampak)
CAN	0018	24	Cancel (tidak tampak)
EM	0019	25	End of medium (tidak tampak)
SUB	001A	26	Substitute (tidak tampak)
ESC	001B	27	Escape (tidak tampak)
FS	001C	28	File separator
GS	001D	29	Group separator
RS	001E	30	Record separator
US	001F	31	Unit separator
SP	0020	32	Spasi
!	0021	33	Tanda seru (exclamation)

"	0022	34	Tanda kutip dua
#	0023	35	Tanda pagar (kres)
$	0024	36	Tanda mata uang dolar
%	0025	37	Tanda persen
&	0026	38	Karakter ampersand (&)
‘	0027	39	Karakter Apostrof
(	0028	40	Tanda kurung buka
)	0029	41	Tanda kurung tutup
*	002A	42	Karakter asterisk (bintang)
+	002B	43	Tanda tambah (plus)
,	002C	44	Karakter koma
-	002D	45	Karakter hyphen (strip)
.	002E	46	Tanda titik
/	002F	47	Garis miring (slash)
0	0030	48	Angka nol
1	0031	49	Angka satu
2	0032	50	Angka dua
3	0033	51	Angka tiga
4	0034	52	Angka empat
5	0035	53	Angka lima
6	0036	54	Angka enam
7	0037	55	Angka tujuh
8	0038	56	Angka delapan
9	0039	57	Angka sembilan
:	003A	58	Tanda titik dua
;	003B	59	Tanda titik koma
<	003C	60	Tanda lebih kecil
=	003D	61	Tanda sama dengan
>	003E	62	Tanda lebih besar
?	003F	63	Tanda tanya
@	0040	64	A keong (@)
A	0041	65	Huruf latin A kapital
B	0042	66	Huruf latin B kapital
C	0043	67	Huruf latin C kapital
D	0044	68	Huruf latin D kapital
E	0045	69	Huruf latin E kapital
F	0046	70	Huruf latin F kapital
G	0047	71	Huruf latin G kapital
H	0048	72	Huruf latin H kapital
I	0049	73	Huruf latin I kapital
J	004A	74	Huruf latin J kapital
K	004B	75	Huruf latin K kapital
L	004C	76	Huruf latin L kapital
M	004D	77	Huruf latin M kapital
N	004E	78	Huruf latin N kapital
O	004F	79	Huruf latin O kapital

P	0050	80	Huruf latin P kapital
Q	0051	81	Huruf latin Q kapital
R	0052	82	Huruf latin R kapital
S	0053	83	Huruf latin S kapital
T	0054	84	Huruf latin T kapital
U	0055	85	Huruf latin U kapital
V	0056	86	Huruf latin V kapital
W	0057	87	Huruf latin W kapital
X	0058	88	Huruf latin X kapital
Y	0059	89	Huruf latin Y kapital
Z	005A	90	Huruf latin Z kapital
[	005B	91	Kurung siku kiri
\	005C	92	Garis miring terbalik (backslash)
]	005D	93	Kurung sikur kanan
^	005E	94	Tanda pangkat
_	005F	95	Garis bawah (underscore)
`	0060	96	Tanda petik satu
a	0061	97	Huruf latin a kecil
b	0062	98	Huruf latin b kecil
c	0063	99	Huruf latin c kecil
d	0064	100	Huruf latin d kecil
e	0065	101	Huruf latin e kecil
f	0066	102	Huruf latin f kecil
g	0067	103	Huruf latin g kecil
h	0068	104	Huruf latin h kecil
i	0069	105	Huruf latin i kecil
j	006A	106	Huruf latin j kecil
k	006B	107	Huruf latin k kecil
l	006C	108	Huruf latin l kecil
m	006D	109	Huruf latin m kecil
n	006E	110	Huruf latin n kecil
o	006F	111	Huruf latin o kecil
p	0070	112	Huruf latin p kecil
q	0071	113	Huruf latin q kecil
r	0072	114	Huruf latin r kecil
s	0073	115	Huruf latin s kecil
t	0074	116	Huruf latin t kecil
u	0075	117	Huruf latin u kecil
v	0076	118	Huruf latin v kecil
w	0077	119	Huruf latin w kecil
x	0078	120	Huruf latin x kecil
y	0079	121	Huruf latin y kecil
z	007A	122	Huruf latin z kecil
{	007B	123	Kurung kurawal buka
¦	007C	124	Garis vertikal (pipa)
}	007D	125	Kurung kurawal tutup

~	007E	126	Karakter gelombang (tilde)
DEL	007F	127	Delete

NO.2

Intruksi perpindahan data

mov — Move = Instruksi mov menyalin item data yang dirujuk oleh operan pertamanya (misal: Konten register, konten memori, atau nilai konstan) ke lokasi yang dirujuk oleh operan keduanya (misal: Register atau memori).

push — Push dalam stack = Instruksi push menempatkan operan ke atas tumpukan perangkat keras yang didukung dalam memori. Secara khusus, dorong ESP pengurangan pertama dengan 4, lalu operasikan operannya ke konten lokasi 32-bit di alamat (% esp). ESP (penunjuk tumpukan) dikurangi oleh push karena tumpukan x86 turun (tumpukan dari alamat tinggi ke alamat yang lebih rendah).

pop — Pop dari stack = Instruksi pop menghapus elemen data 4-byte dari atas tumpukan yang didukung perangkat keras ke dalam operan yang ditentukan (mis. Register atau lokasi memori).

lea — Memuat alamat yang efektif = Instruksi lea menempatkan alamat yang ditentukan oleh operan pertamanya ke dalam register yang ditentukan oleh operan kedua. (Catatan, isi lokasi memori tidak dimuat, hanya alamat efektif dihitung dan ditempatkan ke dalam register)

Intruksi aritmatika dan logika

add — Pertambahan = Instruksi tambah menambahkan bersama dua operan, menyimpan hasilnya di operan kedua.(Catatan, kedua operan mungkin merupakan register,sedangkan jika paling banyak satu operan mungkin merupakan lokasi memori)

sub — Pengurangan = Sub instruksi menyimpan nilai operan kedua hasil dari pengurangan nilai operan pertama dari nilai operan kedua. (Seperti halnya pertambahan)

inc, dec — Peningkatan, penurunan = Instruksi inc menambah konten operan satu. Instruksi des decrements isi operan satu.

imul — Perkalian = Instruksi imul memiliki dua format dasar:

1. Dua-operan (dua daftar sintaks pertama di atas) : Bentuk dua operan melipatgandakan dua operan dan menyimpan hasilnya di operan kedua. Operand hasil (mis. Kedua) harus berupa register.
2. Tiga-operan (dua daftar sintaks terakhir di atas) : Tiga bentuk operan mengalikan operan kedua dan ketiga bersama-sama dan menyimpan hasilnya dalam operan terakhir. Sekali lagi, hasil operan harus berupa register. Selain itu, operan pertama dibatasi menjadi nilai konstan.

idiv — Pembagian = Instruksi idiv membagi konten integer 64 bit EDX: EAX (dibangun dengan melihat EDX sebagai empat byte paling signifikan dan EAX sebagai 4 byte paling tidak signifikan) dengan nilai operan yang ditentukan. Hasil bagi dari divisi disimpan ke dalam EAX, sedangkan sisanya ditempatkan di EDX.

and, or, xor — Logika dan, atau ,dan eksklusif or = Instruksi ini melakukan operasi logis yang ditentukan (logis dan, atau, dan eksklusif atau masing-masing) pada operan mereka, menempatkan hasilnya di lokasi operan pertama.

not — Logika not = Secara logis meniadakan konten operan (yaitu, membalik semua nilai bit di operan).

neg — Negasi = Melakukan negasi komplemen keduanya dari konten operan.

shl, shr — Geser ke kiri, Geser ke kanan = Instruksi-instruksi ini menggeser bit dalam konten operan pertama mereka ke kiri dan kanan, melapisi posisi bit kosong yang dihasilkan dengan nol. Operan yang digeser dapat digeser hingga 31 tempat. Jumlah bit untuk bergeser ditentukan oleh operan kedua, yang bisa berupa konstanta 8-bit atau register CL. Dalam kedua kasus, pergeseran jumlah lebih besar dari 31 dilakukan modulus 32.

Instruksi kontrol aliran

jmp — Jump = Transfer aliran kontrol program ke instruksi di lokasi memori yang ditunjukkan oleh operan.

jcondition — Conditional jump =

je <label> (jump ketika sama)
jne <label> (jump ketika tidak sama)
jz <label> (jump ketika hasil akhir adalah 0)
jg <label> (jump ketika lebih besar dari)
jge <label> (jump ketika lebih besar sama dengan dari)
jl <label> (jump ketika kurang dari)
jle <label> (jump ketika kurang sama dengan dari)

Instruksi ini adalah lompatan bersyarat yang didasarkan pada status serangkaian kode kondisi yang disimpan dalam register khusus yang disebut kata status mesin. Isi kata status mesin mencakup informasi tentang operasi aritmatika terakhir yang dilakukan. Misalnya, satu bit kata ini menunjukkan jika hasil terakhir adalah nol. Lain menunjukkan jika hasil terakhir negatif. Berdasarkan kode kondisi ini, sejumlah lompatan bersyarat dapat dilakukan. Sebagai contoh, instruksi jz melakukan lompatan ke label operan yang ditentukan jika hasil operasi aritmatika terakhir adalah nol. Jika tidak, kontrol dilanjutkan ke instruksi berikutnya secara berurutan.

Sejumlah cabang bersyarat diberikan nama yang secara intuitif didasarkan pada operasi terakhir yang dilakukan menjadi instruksi pembanding khusus, cmp (lihat di bawah). Sebagai contoh, cabang bersyarat seperti jle dan jne didasarkan pada melakukan operasi cmp pertama pada operan yang diinginkan.

cmp — Bandingkan(compare) = Bandingkan nilai dari dua operan yang ditentukan, atur kode kondisi dalam kata status mesin dengan tepat. Instruksi ini setara dengan sub instruksi, kecuali hasil dari pengurangan dibuang alih-alih mengganti operan pertama.

call, ret — Subroutine call(panggilan subrutin) and return = Instruksi ini menerapkan panggilan subrutin dan kembali. Instruksi panggilan pertama-tama mendorong lokasi kode saat ini ke tumpukan perangkat yang didukung dalam memori (lihat instruksi push untuk detail), dan kemudian melakukan lompatan tanpa syarat ke lokasi kode yang ditunjukkan oleh operan label. Berbeda dengan instruksi lompat sederhana, instruksi panggilan menyimpan lokasi untuk kembali ke ketika subrutin selesai.

Instruksi ret mengimplementasikan mekanisme pengembalian subrutin. Instruksi ini pertama-tama mengeluarkan lokasi kode dari perangkat keras yang mendukung tumpukan memori (lihat instruksi pop untuk detailnya). Kemudian melakukan lompatan tanpa syarat ke lokasi kode yang diambil.