**SISTEM OPERASI**

**MODUL 1**

****

**Oleh :**

**NIA ANNISA DAMAYANTI**

**L200190119**

**INFORMATIKA**

**FAKULTAS KOMUNIKASI DAN INFORMATIKA**

**UNIVERSITAS MUHAMMADIYAH SURAKARTA**

**2020**

1. Apa yang dimaksud dengan kode ‘ASCII’, buatlah table kode ASCII lengkap cukup kode ASCII yang standar tidak perlu extended, tuliskan kode ASCII dalam format angka desimal, binary dan hexadesimal serta karakter dan simbol yang dikodekan.

**Jawaban :** ASCII (American Standard Code for Information Interchage) merupakan standar pengkodean karakter dan symbol seperti Unicode dan Hex tetapi ASCII lebih bersifat universal untuk alat komunikasi. Kode ASCII mewakili teks dalam computer, peralatan telekomunikasi dan perangkat lainnya.

**Table Kode ASCII :**

|  |  |  |  |
| --- | --- | --- | --- |
| Desimal | Heksa Desimal | Karakter | Biner |
| 0 | 0000 | NUL | 0000 0000 |
| 1 | 0001 | SOH | 0000 0001 |
| 2 | 0002 | STX | 0000 0010 |
| 3 | 0003 | ETX | 0000 0011 |
| 4 | 0004 | EOT | 0000 0100 |
| 5 | 0005 | ENQ | 0000 0101 |
| 6 | 0006 | ACK | 0000 0110 |
| 7 | 0007 | BEL | 0000 0111 |
| 8 | 0008 | BS | 0000 1000 |
| 9 | 0009 | HT | 0000 1001 |
| 10 | 000A | LF | 0000 1010 |
| 11 | 000B | VT | 0000 1011 |
| 12 | 000C | FF | 0000 1100 |
| 13 | 000D | CR | 0000 1101 |
| 14 | 000E | SO | 0000 1110 |
| 15 | 000F | SI | 0000 1111 |
| 16 | 0010 | DLE | 0001 0000 |
| 17 | 0011 | DC1 | 0001 0001 |
| 18 | 0012 | DC2 | 0001 0010 |
| 19 | 0013 | DC3 | 0001 0011 |
| 20 | 0014 | DC4 | 0001 0100 |
| 21 | 0015 | NAK | 0001 0101 |
| 22 | 0016 | SYN | 0001 0110 |
| 23 | 0017 | ETB | 0001 0111 |
| 24 | 0018 | CAN | 0001 1000 |
| 25 | 0019 | EM | 0001 1001 |
| 26 | 001A | SUB | 0001 1010 |
| 27 | 001B | ESC | 0001 1011 |
| 28 | 001C | FS | 0001 1100 |
| 29 | 001D | GS | 0001 1101 |
| 30 | 001E | RS | 0001 1110 |
| 31 | 001F | US | 0001 1111 |
| 32 | 0020 | spasi | 0010 0000 |
| 33 | 0021 | ! | 0010 0001 |
| 34 | 0022 | “ | 0010 0010 |
| 35 | 0023 | # | 0010 0011 |
| 36 | 0024 | $ | 0010 0100 |
| 37 | 0025 | % | 0010 0101 |
| 38 | 0026 | & | 0010 0110 |
| 39 | 0027 | ‘ | 0010 0111 |
| 40 | 0028 | ( | 0010 1000 |
| 41 | 0029 | ) | 0010 1001 |
| 42 | 002A | \* | 0010 1010 |
| 43 | 002B | + | 0010 1011 |
| 44 | 002C | , | 0010 1100 |
| 45 | 002D | - | 0010 1101 |
| 46 | 002E | . | 0010 1110 |
| 47 | 002F | / | 0010 1111 |
| 48 | 0030 | 0 | 0011 0000 |
| 49 | 0031 | 1 | 0011 0001 |
| 50 | 0032 | 2 | 0011 0010 |
| 51 | 0033 | 3 | 0011 0011 |
| 52 | 0034 | 4 | 0011 0100 |
| 53 | 0035 | 5 | 0011 0101 |
| 54 | 0036 | 6 | 0011 0110 |
| 55 | 0037 | 7 | 0011 0111 |
| 56 | 0038 | 8 | 0011 1000 |
| 57 | 0039 | 9 | 0011 1001 |
| 58 | 003A | : | 0011 1010 |
| 59 | 003B | ; | 0011 1011 |
| 60 | 003C | < | 0011 1100 |
| 61 | 003D | = | 0011 1101 |
| 62 | 003E | > | 0011 1110 |
| 63 | 003F | ? | 0011 1111 |
| 64 | 0040 | @ | 0100 0000 |
| 65 | 0041 | A | 0100 0001 |
| 66 | 0042 | B | 0100 0010 |
| 67 | 0043 | C | 0100 0011 |
| 68 | 0044 | D | 0100 0100 |
| 69 | 0045 | E | 0100 0101 |
| 70 | 0046 | F | 0100 0110 |
| 71 | 0047 | G | 0100 0111 |
| 72 | 0048 | H | 0100 1000 |
| 73 | 0049 | I | 0100 1001 |
| 74 | 004A | J | 0100 1010 |
| 75 | 004B | K | 0100 1011 |
| 76 | 004C | L | 0100 1100 |
| 77 | 004D | M | 0100 1101 |
| 78 | 004E | N | 0100 1110 |
| 79 | 004F | O | 0100 1111 |
| 80 | 0050 | P | 0101 0000 |
| 81 | 0051 | Q | 0101 0001 |
| 82 | 0052 | R | 0101 0010 |
| 83 | 0053 | S | 0101 0011 |
| 84 | 0054 | T | 0101 0100 |
| 85 | 0055 | U | 0101 0101 |
| 86 | 0056 | V | 0101 0110 |
| 87 | 0057 | W | 0101 0111 |
| 88 | 0058 | X | 0101 1000 |
| 89 | 0059 | Y | 0101 1001 |
| 90 | 005A | Z | 0101 1010 |
| 91 | 005B | [ | 0101 1011 |
| 92 | 005C | / | 0101 1100 |
| 93 | 005D | ] | 0101 1101 |
| 94 | 005E | ^ | 0101 1110 |
| 95 | 005F | \_ | 0101 1111 |
| 96 | 0060 | ` | 0110 0000 |
| 97 | 0061 | a | 0110 0001 |
| 98 | 0062 | b | 0110 0010 |
| 99 | 0063 | c | 0110 0011 |
| 100 | 0064 | d | 0110 0100 |
| 101 | 0065 | e | 0110 0101 |
| 102 | 0066 | f | 0110 0110 |
| 103 | 0067 | g | 0110 0111 |
| 104 | 0068 | h | 0110 1000 |
| 105 | 0069 | i | 0110 1001 |
| 106 | 006A | j | 0110 1010 |
| 107 | 006B | k | 0110 1011 |
| 108 | 006C | l | 0110 1100 |
| 109 | 006D | m | 0110 1101 |
| 110 | 006E | n | 0110 1110 |
| 111 | 006F | o | 0110 1111 |
| 112 | 0070 | p | 0111 0000 |
| 113 | 0071 | q | 0111 0001 |
| 114 | 0072 | r | 0111 0010 |
| 115 | 0073 | s | 0111 0011 |
| 116 | 0074 | t | 0111 0100 |
| 117 | 0075 | u | 0111 0101 |
| 118 | 0076 | v | 0111 0110 |
| 119 | 0077 | w | 0111 0111 |
| 120 | 0078 | x | 0111 1000 |
| 121 | 0079 | y | 0111 1001 |
| 122 | 007A | z | 0111 1010 |
| 123 | 007B | { | 0111 1011 |
| 124 | 007C | | | 0111 1100 |
| 125 | 007D | } | 0111 1101 |
| 126 | 007E | ~ | 0111 1110 |
| 127 | 007F | DEL | 0111 1111 |

1. Carilah daftar perintah bahasa assembly untuk mesin intel keluarga x86 lengkap (dari buku referensi atau internet). Daftar perintah ini dapat digunakan sebagai pedoman untuk memahami program ‘boot.asm’ dan ‘kernel.asm’.

**Bahasa Assembly Intel Keluarga x86 :**

* ACALL *(Absolute Call)*
* ADD *(Add Immediate Data)*
* ADDC *(Add Carry Plus Immediate Data to Accumulator)*
* AJMP *(Absolute Jump)*
* ANL *(Logical AND memori ke akumulator)*
* CJNE *(Compare Indirect Address to Immediate Data)*
* CLR *(Clear Accumulator)*
* CPL *(Complement Accumulator)*
* DA *(Decimal Adjust Accumulator)*
* DEC *(Decrement Indirect Address)*
* DIV *(Divide Accumulator by B)*
* DJNZ *(Decrement Register And Jump Id Not Zero)*
* INC *(Increment Indirect Address)*
* JB *(Jump if Bit is Set)*
* JBC *(Jump if Bit Set and Clear Bit)*
* JC *(Jump if Carry is Set)*
* JMP *(Jump to sum of Accumulator and Data Pointer)*
* JNB *(Jump if Bit is Not Set)*
* JNC *(Jump if Carry Not Set)*
* JNZ *(Jump if Accumulator Not Zero)*
* JZ *( Jump if Accumulator is Zero )*
* LCALL *( Long Call )*
* LJMP *( Long Jump )*
* MOV *( Move From Memory )*
* MOVC *( Move From Codec Memory )*
* MOVX *(Move Accumulator to External Memory Addressed by Data Pointer)*
* MUL *( Multiply )*
* NOP *( No Operation )*
* ORL *(Logical OR Immediate Data to Accumulator)*
* POP *(Pop Stack to Memory)*
* PUSH *(Push Memory onto Stack)*
* RET *(Return from subroutine)*
* RETI *( Return From Interrupt )*
* RL *(Rotate Accumulator Left)*
* RLC *( Rotate Left through Carry )*
* RR *( Rotate Right )*
* RRC *( Rotate Right through Carry )*
* SETB *(set Carry flag)*
* SJMP *(Short Jump)*
* SUBB *( Subtract With Borrow )*
* SWAP *( Swap Nibbles )*
* XCH *( Exchange Bytes )*
* XCHD *( Exchange Digits )*
* XRL *( Exclusive OR Logic )*