

MODUL INSTRUKSI BAHASA ASSEMBLY

A. Kompetensi Dasar

Menyajikan instruksi bahasa assembly mikroprosesor

B. Tujuan

1. Siswa dapat mengetahui bahasa assembly.
2. Siswa dapat memahami instruksi bahasa *assembly*.
3. Siswa dapat memahami urutan penggunaan instruksi bahasa assembly.

C. Dasar Teori

1. Bahasa Rakitan dan Posisinya dengan Bahasa Lain.

Manusia dapat melakukan interaksi secara efektif dengan menggunakan media bahasa, sehingga memungkinkan untuk menyampaikan gagasan, pemikiran dan informasi, tanpa ada bahasa maka komunikasi tidak akan terjadi. Dalam lingkungan komputer bahasa pemrograman bertindak sebagai sarana komunikasi antara manusia dan komputer dalam menyelesaikan permasalahannya, tanpa ada bahasa pemrograman maka komputer hanyalah sebagai suatu alat saja yang tidak dapat membantu manusia. Suatu solusi untuk suatu masalah akan menjadi lebih mudah bila bahasa pemrograman yang digunakan lebih dekat dengan permasalahan tersebut. Oleh karena itu, bahasa harus memiliki konstruksi yang merefleksikan terminologi dan elemen yang dipergunakan dalam mendeskripsikan masalah dan bahasa tersebut bebas terhadap jenis komputer yang digunakan. Bahasa pemrograman yang dapat merefleksikan terminologi tersebut adalah bahasa pemrograman yang lebih menyerupai bahasa manusia sendiri dan dikenal dengan bahasa tingkat tinggi (High Level Language). Disisi lain, komputer digital melaksanakan pekerjaannya dalam memecahkan masalah manusia hanya memahami dan menerima bahasa yang berupa nol dan satu atau yang dikenal sebagai bahasa tingkat rendah (Low Level Language).

Bahasa pemrograman berdasarkan ketergantungannya dengan mesin bisa meliputi jenis bahasa seperti berikut ini:

1. Bahasa Mesin, yang merupakan bentuk bahasa terendah dari bahasa komputer, dimana setiap instruksi dalam program direpresentasikan dengan kode numerik, yang secara fisik berupa deretan angka 0 dan 1. Sekumpulan instruksi dalam bahasa mesin bisa dibentuk menjadi microcode, yaitu semacam prosedur dalam bahasa mesin.

2. Bahasa Rakitan / Assembly Language, yang merupakan bentuk simbolik dari bahasa mesin. Misalnya simbolik / mnemonic MOV AH,2 adalah terjemahan dari bahasa mesin 1011010000000010. Pada bahasa rakitan terdapat alat bantu untuk diagnostik yang disebut debug. Contoh produk yang ada untuk pengembangan dan debug bahasa rakitan adalah : Turbo Assembler dari Borland Co. dan Macro Assembler dari Microsoft Co, sedangkan debuggernya adalah Turbo Debugger dan Debug. Ciri khas bahasa rakitan dalam penulisan perintahnya adalah terdiri dari field operasi atau perintah diikuti oleh satu atau lebih operannya.
3. Bahasa Tingkat Tinggi (user oriented), disebut demikian karena bahasa pemrograman komputer ini lebih mendekati dengan bahasa manusia. Disamping itu memberikan fasilitas yang lebih banyak, kontrol program yang terstruktur, kalang (nested), block dan prosedur. Contohnya adalah bahasa Pascal, BASIC, FORTRAN, COBOL, ADA, C++, dan lain-lain.
4. Bahasa Problem Oriented atau Object Oriented, yang memungkinkan penyelesaian untuk suatu masalah atau aplikasi yang spesifik. Contohnya SQL untuk database, LISP untuk gambar teknik, Visual Basic, Visual Delphi, COGO untuk aplikasi teknik sipil, Flash untuk pembuatan animasi, dan lain-lain.

Muara akhir dari kelompok bahasa – bahasa tadi adalah ke bahasa mesin atau bahasa rakitan, dimana ketika telah dikompilasi maka hasilnya akan menjadi kode-kode dalam bahasa mesin atau mnemonic assembler. Perhatikan instruksi biner berikut :

1011010000000010	MOV AH,2	}	PRINT "A"
1011001001000001	MOV DL,41		
1100110100100001	INT 21		
1100110100100000	INT 20		
			END

Pada baris pertama instruksi bilangan biner tersebut, memerintahkan komputer untuk mempersiapkan service 2 untuk persiapan proses cetak dari interrupt 21 yang diinstruksikan pada baris ke 3. Baris keduanya menunjukkan pada komputer karakter yang akan dicetak adalah karakter 41 Heksa atau 65 Desimal yang artinya huruf A, sehingga waktu komputer mengerjakan perintah baris ketiga, tercetaklah huruf A di layar. Baris terakhir memberikan perintah pada komputer untuk menghentikan program.

Bilangan pada bahasa rakitan dapat berupa beberapa basis bilangan, yaitu : bilangan desimal, heksadesimal, oktal, dan biner. Tetapi yang paling banyak digunakan pada pemrograman bahasa rakitan adalah bilangan heksadesimal. Bilangan desimal : bilangan

yang berbasis 10 yang biasa kita gunakan sehari-hari. Bilangan ini dipakai menghitung dari mulai 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Bilangan biner adalah bilangan yang berbasis 2, dimana perhitungan pada bilangan ini hanya dikenal dua bentuk bilangan 0 dan 1 saja. Bilangan oktal adalah bilangan yang berbasis delapan, sehingga dalam perhitungan hanya dikenal 8 bilangan. Dengan demikian bila kita menghitung mulai dari 0 hingga 15 dalam bilangan biner, oktal dan heksadesimal menjadi seperti tabel berikut:

Desimal	Binner	Oktal	Heksadesimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Konversi bilangan Biner menjadi Desimal :

Contoh : bilangan biner 110 maka desimalnya adalah :

$$\begin{aligned}
 &= 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \\
 &= 4 + 2 + 0 \\
 &= 6_{10}
 \end{aligned}$$

Konversi bilangan Desimal menjadi Biner :

Contoh : bilangan desimal 13 maka binernya :

$$\begin{array}{rcl}
 2 \overline{) 13} & \text{sisa } 1 & \\
 2 \overline{) 6} & \text{sisa } 0 & \\
 2 \overline{) 3} & \text{sisa } 1 & \\
 2 \overline{) 1} & \text{sisa } 1 & \\
 0 & &
 \end{array}$$

↑
baca binnernya

Jadi bilangan binernya adalah 1101

Konversi bilangan Oktal ke Desimal

Contoh bilangan 23 oktal ke desimal :

$$\begin{aligned}
 &= 2 \times 8^1 + 3 \times 8^0 \\
 &= 16 + 3 \\
 &= 19 \text{ Desimal}
 \end{aligned}$$

Konversi bilangan Desimal ke Oktal

Contoh bilangan 50 desimal ke oktal :

$$\begin{array}{rcl}
 {}_8 50 & \text{sisa } 2 & \uparrow \\
 {}_8 6 & \text{sisa } 6 & \uparrow \\
 0 & &
 \end{array}$$

Konversi Heksadesimal ke Biner

Contoh bilangan C5E2 heksadesimal menjadi biner :

$$C \ 5 \ E \ 2 = 1100 \ 0101 \ 1110 \ 0010$$

Jadi bilangan binernya = 1100010111100010

Untuk konversi ke bilangan desimal lakukan seperti dari Biner ke Desimal.

2. Instruksi

Instruksi atau perintah dalam bahasa assembler terdiri dari susunan kode biner yang membentuk tata cara dan tata kelola operasi hardware mikroprosesor, kode biner digunakan sebagai dasar pembentuk instruksi dengan alasan bahwa sebuah mikroprosesor secara hardware hanya mengenali dua kondisi yaitu nol dan satu.

Contoh penulisan instruksi:

Instruksi	Siklus	Op-code	Ukuran
ADC A _r (HL)	7	8E	1
ADC A _r (IX+o)	19	DD 8E oo	3
ADC A _r (IY+o)	19	FD 8E oo	3
ADC A _r n	7	CE nn	2
ADC A _r r	4	88+r	1
ADC A _r [Xp]	8	DD 88+p	2
ADC A _r [Yq]	8	FD 88+q	2
ADC HL,BC	15	ED 4A	2
ADC HL,DE	15	ED 5A	2
ADC HL,HL	15	ED 6A	2
ADC HL,SP	15	ED 7A	2
ADD A _r (HL)	7	86	1
ADD A _r (IX+o)	19	DD 86 oo	3
ADD A _r (IY+o)	19	FD 86 oo	3
ADD A _r n	7	C6 nn	2
ADD A _r r	4	80+r	1
ADD A _r [Xp]	8	DD 80+p	2
ADD A _r [Yq]	8	FD 80+q	2
ADD HL,BC	11	09	1
ADD HL,DE	11	19	1
ADD HL,HL	11	29	1

B. Pengelompokan Instruksi Mikroprosesor Berdasarkan Fungsinya

➤ **Instruksi mengisi dan menukar**

Instruksi mengisi dan menukar isi dari register dan atau memori dibutuhkan pengalamatan yang diikuti sertakan ke dalam setiap instruksi, alamat tersebut digunakan untuk menunjuk sumber data dan tujuan data yang diinginkan.

Sebagai contoh, mengisi register C dari register B menggunakan op-code 48H, dalam bilangan biner adalah 0100 1000. sedangkan mnemonic bahasa assembly untuk seluruh kelompok ini adalah LD, diikuti dengan tujuan dan sumber.

➤ **Transfer blok**

Instruksi transfer blok yang sangat baik untuk memindahkan sejumlah data, instruksi beroperasi dengan tiga register, yaitu:

1. HL penunjuk lokasi sumber
2. DE penunjuk lokasi tujuan
3. BC adalah byte counter

➤ **Arithmatika dan logika**

ALU, singkatan dari *Arithmetic And Logic Unit* (unit aritmatika dan logika), adalah bagian mikroprosesor yang berfungsi melakukan operasi hitungan aritmatika dan logika, operasi aritmatika meliputi operasi penjumlahan dan pengurangan. Sedangkan operasi logika yang dapat dilakukan oleh ALU meliputi logika AND dan OR.

Sebagai contoh untuk instruksi DEC atau instruksi INC yang merupakan instruksi beroperasi dengan 8-bit, dan operasinya dilakukan antara data yang berada di akumulator dan sumber data serta hasil operasi ditempatkan di akumulator.

➤ **Putar dan geser**

instruksi putar dan geser pada mikroprosesor Z80 adalah untuk memutar atau menggeser data dalam akumulator, atau data yang berada dalam setiap register general-purpose, atau lokasi memori.

➤ **Bit manipulasi(Set, Reset, test)**

Manipulasi bit memungkinkan untuk mengubah isi bit dalam register flag, yaitu dengan tujuan fasilitasi perangkat lunak untuk pemanggilan rutin tertentu, indikasi kontrol kondisi eksternal, atau data dikemas ke lokasi memori, membuat pemanfaatan memori yang lebih efisien. Instruksi manipulasi bit dalam mikroprosesor Z80 dapat men-set, reset, atau menguji bit dalam akumulator, setiap register umum atau lokasi memori melalui sebuah instruksi

tunggal.

➤ **Jump, Call, dan Return**

Instruksi ini berfungsi untuk melayani permintaan program untuk melakukan lompat dari satu alamat ke alamat tertentu, panggilan untuk sebuah rutin, dan kembali ke program utama saat rutin yang dipanggil telah selesai tugasnya. Sebagai contoh, sebuah lompatan tanpa syarat ke lokasi memori 3E32H adalah:

➤ **Input/Output**

Mikroprosesor Z80 memiliki serangkaian instruksi input dan output, pengalamatan dari perangkat input atau output dapat berupa absolut atau langsung, menggunakan register C. Dalam register mode pengalamatan tidak langsung, data dapat ditransfer antara perangkat I/O dan salah satu register internal. Selain itu delapan blok instruksi transfer telah dilaksanakan, instruksi-instruksi ini mirip dengan transfer blok memori.

➤ **Dasar Kontrol CPU**

Beberapa tujuan secara umum instruksi kontrol CPU, meliputi instruksi NOP merupakan instruksi yang tidak melakukan apa-apa, Instruksi HALT menghentikan CPU sementara, DI dan EI adalah operasi interupsi yang diterima digunakan untuk mengunci atau mengaktifkan interupsi.

D. Soal

1. Sebutkan macam-macam instruksi mikroprosesor berdasarkan fungsinya dan jelaskan!
2. Ubah bilangan berikut kedalam bentuk desimal
 - a. 1101
 - b. 23
 - c. C5E2