

BAHASA PEMROGRAMAN MIKROKOMPUTER

BAHASA MESIN

PROGRAM BAHASA MESIN

Sebuah program diperlukan saat mikrokomputer digunakan untuk melaksanakan sederetan operasi (menghitung, memasukkan dan mengeluarkan data) seperti halnya dalam proses pengontrolan alat-alat. Program tersebut tersusun dari sejumlah perintah yang berurutan yang dibaca dan dieksekusi.

Namun sebuah mikroprosesor hanya mampu membaca bahasa mesin saja. Di dalam program seperti BASIC, FORTRAN, yang dikenal sebagai bahasa tingkat tinggi, sudah ada bagian yang berfungsi menterjemahkannya ke bahasa mesin. Sehingga pemakai lebih mudah memahami bahasa BASIC/Fortrain. Sebelum dieksekusi program tersebut akan diterjemahkan dahulu ke bahasa mesin, sehingga prosesnya menjadi lebih lambat (beberapa ratus kali lebih lambat). Karena pemakaian bahasa tingkat tinggi akan memperlambat suatu proses, karena itulah dipilih bahasa mesin untuk mengontrol mesin. Panjang word suatu perintah diset dalam satuan 8 bit(=1byte). Bergantung kepada jumlah bit yang digunakan, ada empat jenis perintah; perintah 1-byte, perintah 2-byte, 3-byte dan perintah 4-byte.

Pernyataan Perintah

Bahasa mesin mk disajikan oleh kombinasi “1” dan “0”. Pada CPU 8-bit perintah (instruksi) dinyatakan dengan menggabungkan beberapa kelompok bil biner yang terdiri dari 8 digit.

Tapi bekerja dengan bahasa mesin akan banyak kesulitan karena orang sangat sukar untuk mengingatnya, karena itu dipecah menjadi 2 bagian (4 bit, 4 bit) yang kemudian disajikan dalam kode bilangan hexadecimal.

Contohnya bahasa mesin diatas dinyatakan dalam kode hexa sebagai berikut:

Namun inipun masih dirasakan sulit bagi orang untuk memahaminya. Maka digunakan pernyataan yang dikenal dengan kode mnemonic, yaitu kumpulan kata-kata yang terdiri beberapa huruf, yang merupakan kependekan perintah-perintah. Satu CPU dengan CPU lainnya mempunyai kode mnemonic yang berbeda. Contoh nya pada Z-80

LD A,D

LD berarti Load Data dan disebut sebuah *kode perintah* karena berkaitan dengan isi perintah. A,D disebut *operand*, yaitu yang menjadi objek perintahnya.

Tabel berikut menunjukkan contoh-contoh perintah :

	Mnemonic	Kode mesin		Kode HExa
Perintah 2-byte	ADD A, 05H	1100	0110	C6
		0000	0101	05
Perintah 3-byte	JP 8031H	1100	0011	C3
		0011	0001	31

		1000	0000	80
Perintah 4-byte	LD BC, (9010H)	1110	1101	ED
		0100	1011	4B
		0001	0000	10
		1001	0000	90

Dasar-dasar Pemrograman

Karena mk bekerja untuk tujuan tertentu, beberapa perintah harus dilink dalam urutan yang sesuai. Sebuah program terdiri dari sekumpulan perintah yang dieksekusi secara berurutan dan jika tidak ada kesalahan, akan bekerja sesuai dengan keinginan.

Diagram Alur

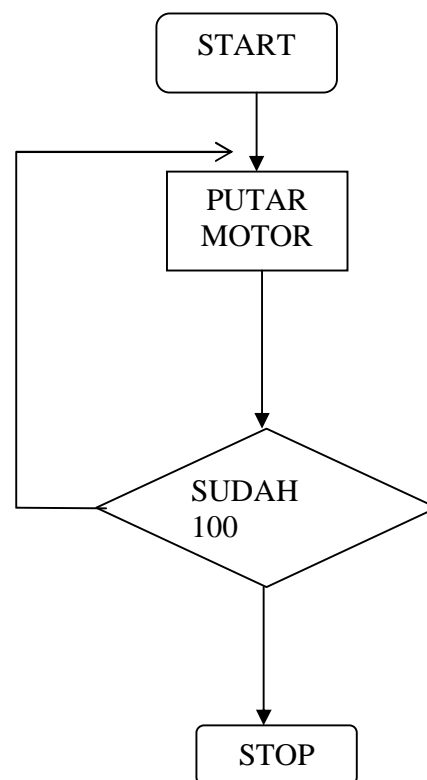
Sekali tujuan sudah ditetapkan, maka dapat dibuat programnya, tapi belum tentu berhasil apabila ada kesalahan-kesalahan. Ada baiknya sebelum menuliskan program membuat diagram alur terlebih dahulu.

Contohnya . Tujuan program kita adalah “memutar motor 100 kali” Maka sebenarnya ada dua kegiatan yaitu “memutar motor” dan “Mencek apakah sudah berputar 100 kali” Maka dalam diagram alur dilukiskan sbb:

Diagram alur ini sangat penting khususnya untuk program-program yang cukup kompleks, sehingga akan memudahkan dalam pencarian kesalahan, dan memperpendek waktu.

Simbol-simbol dan panah yang digunakan dalam diagram alur ini sudah standar mengikuti JIS (Japanish Industrial Standard)

Adapun simbol-simbol penting ditunjukkan dibawah ini



Pengkodean

Jika pembuatan diagram alur sudah selesai, selanjutnya adalah menterjemahkannya ke perintah-perintah yang dipahami oleh mikrokomputer, sesuai dengan alur tadi. Pertama, tulis dulu dalam kode mnemonic (bahasa assembler).

Pertama tulis perintah-perintah dalam operation code (kode perintah) berikut operandnya dengan pernyataan mnemonic. Bersamaan dengan itu, pernyataan lain (komentar) yang diperlukan disertakan, untuk membedakan dengan perintah sejenis lainnya. Khususnya untuk perintah JP, alamat kemana lompatnya harus disertakan. Selanjutnya adalah tuliskan alamat awal dimana program itu akan dimulai. Contohnya ADDRESS 8000H. Terakhir perintah-perintah tadi sesuai alamat di masukkan kedalam kode mesin.

Address	Kode Mesin		Label	Op.Code	Operand	Comments
				ORG	8000H	
8000	DB	FC	INPUT	IN	A, (0FCH)	
8002	C6	02		ADD	A, 02H	;
8004	D3	FD		OUT	(0FDH), A	
8006	C3	00	80	JP	INPUT	;

Kode mesin
Mnemonic

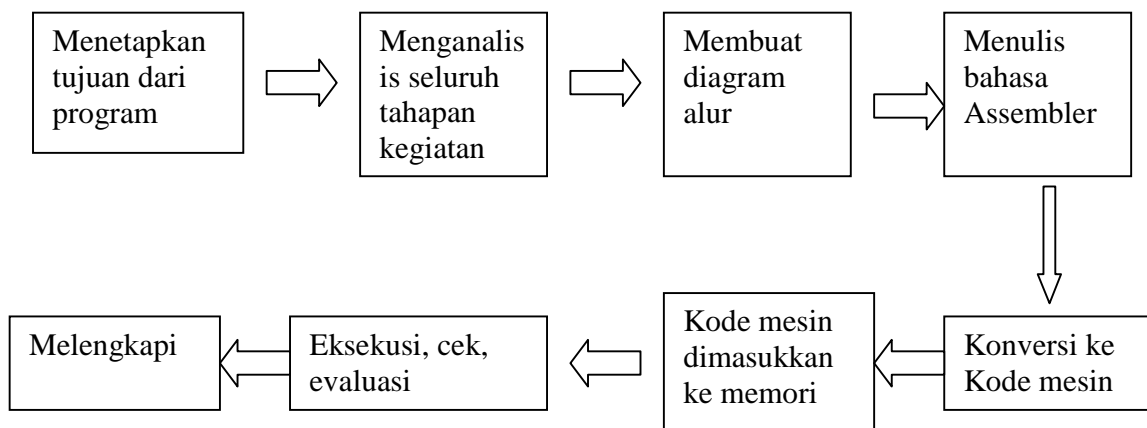
Sebagai contoh, IN A, (0FCH) adalah perintah yang melibatkan DB dan FC. Karena kode mesin ini adalah 2-byte, maka alamat untuk pernyataan berikutnya akan 8002H.

Sehingga seluruh program ini baru komplit setelah mengulang kembali proses.

Apakah adanya alamat jump dari perintah JP yang merubah label? Ini penting mengingat akan mengganti pernyataan alamat (dalam contoh ini 8000H) sebagai alamat jump. Dengan langkah seperti ini lah (yaitu mentransfer dari perintah dalam mnemonic ke kode mesin dan kemudian menset alamatnya baru program keseluruhan siap di eksekusi..

Catatan:

Saat menset alamat heksa dalam 4 digit, seperti misalnya alamat jump ke kode mesin, urutan 2 bit atas dan 2 bit bawah adalah terbalik. (Misal 8000H disusun menjadi 0080)



Menulis program dengan bahasa Assembler

Bahasa assembler cukup banyak dan perbedaan satu dengan lainnya sangat sedikit antar satu perusahaan dengan lainnya. Bahasa Assembler yang dijelaskan disini adalah yang digunakan untuk Z-80 dari perusahaan Zilog.

Label	Opecode	Operand	Commant
	ORG	8000H	
	LD	A, 11H	
LOOP:	INC	A	
	OUT	(0FDH),A	
	JP	LOOP	
	END		

Pada tabel diatas banyak kode mesin, yang akan dikonversi ke bahasa mesin.

Ada empat bagian dari seluruh program tersebut:

1. Kolom Label Menggantikan alamat dari perintah (mirip label nama)
2. Kolom Kode Operation Menggambarkan jenis perintahnya (transfer, add)
3. Kolom Operand Tujuan/sasaran perintah (register, port FD dll)
4. Kolom komentar Penjelasan singkat program/perintah

Merancang Alamat

Dalam bahasa Assembler, penting dalam menetapkan alamat memori dalam kode mesin. Misalnya ORG8000H menunjukkan bahwa program akan dimulai dari alamat 8000H.

Pernyataan bilangan.

Ada tiga jenis sistem bilangan yang bisanya dipakai dalam Kode biner, bilangan desimal dan bilangan Heksadecimal

Pernyataan perintah

Label

Label ini biasanya digunakan untuk menunjuk alamat jump. Contohnya pada label loop[menunjukkan alamat instruksi INC (kode mesinnya adalah 3C)

Operan dari JP saat diterjemahkan ke kode msin harus diset ke alamat INC

Contohnya alamat label 8002H sebagai Loop JP diterjemahkan menjadi C3 02 80 dalam bahasa mesin

Nama label ada yang..

Merupakan kombinasi huruf dan angka yang terdiri dari sekitar 6 karakter, misalnya START, REFTO1 dll.

Tidak digunakan kode mnemonik aatau konde instruksi(misalnya LD, ORG tidak boleh digunakan untuk label)

Kolon (:) harus ditambahkan diujung suatu label

Harus ada spasi antara label dengan opcode

OP CODE

Op Code Adalah isi dari perintah-perintah (seperti LD; ADD dll)

Operand

Operand berkaitan dengan tujuan perintah/ operasi yang dilakukan. (seperti Nama-nama Register, data atau alamat suatu label atau nilai numeriknya.

Comment

Tempat menuliskan keterangan-keterangan tambahan agar program lebih mudah dimengerti.

Perintah Pseudo

Biasanya digunakan untuk mempermudah membaca program-program yang kompleks.

Bentuk pernyataan pseudo adalah sebagai berikut:

Nama Label (C) operand instruksi pseudo: cp,,emt

Beberapa label instruksi pseudo menunjukkan alamat , kadang pula bukan. Hanya label yang menunjukkan alamat yang diikuti kolon (Ⓢ)

Kode Instruksi pseudo dan maknanya:

ORG

Ini digunakan untuk menunjukkan alamat pertama (asal). Jika tidak dilakukan operasi ini, maka program akan dimulai dari alamat 0.

END

Perintah ini menunjukkan akhir suatu program

EQU

Merupakan Nilai bilangan dari label yang ada disebelah kirinya.

Label	Kode	Operand
MOTOR	EQU	2

:

OUT (MOTOR),A

OUT (MOTOR), A berarti mengirimkan data dari akumulator ke port output yang dihubungkan ke rangkaian untuk menggerakkan motor. Disini, OPERAND MOTOR merujuk ke nomor port keluaran. Pada baris intruksi pertama dinyatakan MOTOR=2, berarti MOTOR adalah port keluaran nomor 2. Perintah OUT (MOTOR),A biasanya tampil berulang kali dalam program, tapi dengan nomor port yang berganti-ganti.

Jadi dalam bahasa mesin maka perintah OUT MOTOR tadi akan ditulis D3 02, karena dalam kode mesin perintah OUT kodenya D3.

DB (Define byte, mendefinisikan byte)

Perintah ini menyuruh meload data berdasar byte-byte ke memori.

Contoh

Label	Code	Operand	Commend
VAL	EQU	6	
DB1:	DB	VAL	; dikode sebagai 06
	:		
	:		
X:	DB	100	; diberi kode 64H
	DB	102	;diberi kode 66H
	DB	104	;diberi kode 68H
	DB	106	;diberi kode 6AH

Pada baris kedua, nilai VAL adalah semua merujuk ke alamat DB1, dan pada baris pertama VAL didefinisikan 6. Address X dinyatakan dalam desimal 100. Baris berikutnya adalah alamat X+1 yang mana diisi oleh 102 (=66).

DW (Define Word)

DW berarti word terdefinisi dan perintah ini menyuruh meload data 2 byte ke memori. Tapi byte data yang diset mendahului byte rendah.

Contoh

Label	Code	Operand
YDATE	DW	1983H

83H dis et di alamat YDATE sedang 19H di alamat YDATE+1

DS (Define Storage)

DS berarti mendefinisikan tempat dan perintah ini bermakna memesan tempat dalam di memori sejumlah byte sesuai yang ditunjukkan dalam operandnya.

Contoh

Label	Kode	Operand	Makna
HERE:	DS	10	; Pesan 10 byte kedepan
	DS	16H	;Pesan 16 byte kedepan

DS ini adalah instruksi yang biasanya digunakan untuk memesan area memori dengan jumlah beberapa byte.

Dalam contoh diatas, 26 byte (10 + 16) dipesan, memori tersebut bebas dikirim dan menerima data seperti halnya register.

Program dibawah ini adalah contoh dimana digunakan instruksi pseudo

