

Bab 1

Pemrograman dan Bahasa Rakitan

1.1 Pendahuluan

Komputer dalam melaksanakan tugasnya tidak dapat dipisahkan dari peralatan lainnya yang mendukung kegiatan kerja komputer secara menyeluruh. Di samping terdapat piranti pendukung yang dapat memberi kemajuan bagi komputer, juga memberi ketepatan kerja yang sangat baik. Perangkat yang paling penting bagi komputer dalam melaksanakan tugasnya adalah tersedianya pernyataan yang memberi perintah untuk menyelenggarakan perhitungan dan pengendalian, karena tanpa hal itu komputer tidak akan mampu melakukan apa-apa. Susunan pernyataan ini dalam suatu kumpulan dinyatakan sebagai program.

Pemrograman merupakan cara dengan mana sekumpulan perintah yang dihasilkan bagi sebuah komputer menyelenggarakan aktivitas yang ditentukan. Aktivitas dapat berupa apa saja, tetapi secara umum berkaitan dengan penyelesaian perhitungan atau pengendalian. Komputer tidak dapat diharapkan menyelenggarakan sembarang tugas secara memadai tanpa masalah itu diselesaikan dan dirinci secara benar ke dalam perintah-perintah yang sesuai, yang pada dasarnya mempertanyakan rincian lengkap setiap langkah penyelesaian. Secara umum pembuatan suatu program bergantung kepada empat hal, yaitu:

- a. Memahami masalah yang dihadapi.
- b. Memahami bahasa yang digunakan untuk menyelesaikan masalah.
- c. Memahami menggunakan alat untuk mendapatkan program yang benar.
- d. Pengujian atau pencarian kesalahan.

1.2 Pendefinisian Masalah

Mengerti tentang masalah yang dihadapi merupakan hal penting yang mendasar. Langkah pertama dalam menulis program adalah dengan memikirkan secara hati-hati tentang apa yang diinginkan

dari pelaksanaan program itu. Dengan kata lain, penentuan fungsi dari keseluruhan struktur program dan data yang diperlukan sebaik penentuan algoritma untuk pelaksanaan fungsi-fungsi yang diperlukan.

Secara umum, jika terdapat suatu persoalan yang akan diselesaikan dengan bantuan komputer dan bahasa pemrograman, maka persoalan itu akan diuraikan ke dalam

- a. Hal-hal yang berkaitan dengan masukan, yaitu data dan pernyataan masukan.
- b. Hal-hal yang berkaitan dengan keluaran, yaitu informasi dan pernyataan keluaran.
- c. Hal-hal yang berkaitan dengan proses, yaitu rumus, perulangan, dan kendali bersyarat.

Pada umum ketiga hal di atas, disusun dalam suatu algoritma atau diagram alir. Misalnya, apabila ingin disajikan suatu kalimat:

”Program pertama dengan bahasa rakitan.”

dengan menggunakan bahasa rakitan, maka dapat saja disajikan dengan barisan perintah yang bersifat berurutan (*sequential*), atau susunan perintah dengan menggunakan perulangan maupun struktur bersyarat, dan ini semua tergantung kepada pemahaman dan tujuan penggunaan.

1.3 Penyandian

Penyandian (*coding*) program atau mengikuti kelengkapan diagram alir, mempersiapkan perintah-perintah dalam bentuk tersandi. Langkah-langkah logis yang dinyatakan dalam diagram alir diterjemahkan ke dalam perintah-perintah komputer. Pilihan bahasa tergantung atas kerumitan masalah dan kapasitas dan sumber program yang digunakan.

Bahasa pemrograman rakitan disusun dari beberapa bagian, yaitu: judul (kulit program), deklarasi dan data, tubuh program, komentar dan penutup. Semuanya itu terdiri dari apa yang disebut dengan pengarah (*directive*), operator, simbol pilihan, mnemonic, register, dan lainnya. Misalnya,

Program 1.1: PERTAMA.ASM

```

;-----
; Nama Berkas : 11.ASM
; Pemrogram   : Mahyuddin
;-----
%Title  "Judul - Tidak dirakit"
        IDEAL
        DOSSEG
        MODEL TINY
; ----- Ditempatkan tetapan di sini
        DATASEG
; ----- Deklarasi peubah dan nilainya
Pesan   DB 'Program pertama dengan bahasa'
        DB 'rakitan',13,10,'$'
; ----- Penentuan alamat awal model berkas
        CODESEG
        ORG 100h   ; Alamat berkas bermarga .COM
; ----- Tubuh program atau subrutin
Mulai : MOV AH,09h
        MOV DX,OFFSET Pesan
        INT 21h   ; Sela tampilan Sistem Operasi
                ; (DOS dan Window)
; ----- Akhir sebuah program
        INT 20h   ; Sela SO kembali ke SO
        END Mulai

```

Judul (*header*) akan memulai suatu program bahasa rakitan. Pada judul dijumpai bermacam aba-aba (*command*) dan pengarah, tidak satupun darinya akan dirakit menjadi bahasa mesin, tetapi fungsinya secara umum akan sesuai dengan bagaimana suatu program dihasilkan. Pada program di atas, %Title menyatakan maksud dari program sebagai judul, pengarah MODEL TINY dan pengarah IDEAL menyatakan modus program yang dipilih yang hanya berlaku pada Turbo Assembler (TASM), dan dapat diubah menjadi Program 1.2.

Program 1.2: PERTAMB.ASM

```

;-----
; Nama Berkas : 12.ASM
; Pemrogram   : Mahyuddin
;-----
PengSandi SEGMENT
        ASSUME CS:PengSandi
        ORG 100h
Mulai   : JMP Awal
Pesan   DB 'Program pertama'
        DB 'dengan bahasa rakitan',13,10,'$'
Awal    : MOV AH,09h
        MOV DX,OFFSET Pesan
        INT 21h
        INT 20h

```

```
PengSandi ENDS
          END Mulai
```

Fungsi beberapa pengarah yang digunakan pada program di atas dapat dijelaskan sebagai berikut:

ORG singkatan dari *originate* untuk menentukan alamat awal suatu blok dalam program baik untuk blok data maupun blok perintah.

ASSUME untuk memberikan aba-aba bagi pengarah untuk menggunakan penggal tertentu sebagai penggal data atau program.

SEGMENT untuk menyatakan awal penggal.

ENDS atau END untuk menunjukkan akhir suatu penggal atau akhir program.

DB singkatan dari *define byte* digunakan untuk menyatakan panjang data dan harganya.

OFFSET untuk menerangkan bahwa yang dimaksudkan adalah pergeseran alamat suatu peubah terhadap penggalnya.

Tubuh program dapat saja mengandung label, mnemonik, operan, komentar dan pengarah. Secara umum susunan perintah pada tubuh program dalam satu baris adalah sebagai berikut:

```
Label : MNEMONIK Operan ; Komentar
```

Label akan dinyatakan apabila baris program akan menjadi tempat lompatan dari baris program lain. Label dituliskan sebagaimana a-

danya suatu pengenalan dan dapat dinyatakan sesuai maksud lompatan dan diakhiri dengan titik dua (:).

Mnemonik adalah sekumpulan huruf yang terdiri dari dua sampai lebih, singkatan yang mewakili dari maksud perintah dinyatakan, misalnya MOV disingkat dari MOVE yang berarti pemindahan nilai, yaitu MOV AH,09h berarti AH=09h. Mnemonik dapat saja memerlukan operan, dan kadangkala lebih dari dua operan.

Mnemonik berlaku khusus, seperti INT disingkat dari *interrupt* yang diartikan sela, merupakan pemanggilan perangkat keras yang dihasilkan secara eksternal dari sinyal perangkat keras atau pemanggilan perangkat lunak secara internal dari pelaksanaan perintah atau dengan peristiwa lain. INT berasal dari BIOS (*basic input-output system*) dan SO (sistem operasi), misalnya INT 21h dan INT 20h berasal dari SO yang masing-masing berfungsi dan mempunyai ketentuan sendiri, yaitu

INT 21h Layanan 09h

Fungsi : Untuk menampilkan untaian karakter yang diakhiri \$.

Masukan : AH = 09h
DS:DX = alamat untaian

INT 20h

Fungsi : Untuk kembali ke *prompt* sistem operasi

Masukan : -

Operan mengandung nilai-nilai dengan mana pengoperasian perintah mnemonik didahulukan. Sedikit saja perintah yang tidak memerlukan operan, dan kebanyakan pula perintah memerlukan dua operan. Jika demikian, operan pertama disebut tujuan (*destination*), sedangkan operan kedua (jika ada) disebut sumber (*source*). Operan mengambil banyak bentuk yang berbeda sesuai dengan kebutuhan dan syarat-syarat penggunaannya, misalnya MOV memerlukan dua operan, yaitu AH dan 09h, yang tidak lain berupa salah satu register dan data (nilai) langsung. Sedangkan MOV DX, OFFSET Pesan berlaku cara berbeda, dengan mana operan sumber menggunakan pengarah untuk menunjuk alamat untaian karakter yang diwakili oleh peubah Pesan.

Register adalah salah satu operan, yang berasal dari sistem komputer. Untuk melakukan suatu pekerjaan, pengolah mikro (*microprocessor*) menggunakan register sebagai tempat pengolahannya. Register dikategorikan menjadi dua bagian: register serbaguna dan register khusus.

Register serba guna (*general-purpose register*) atau disingkat register umum terdiri dari empat register yaitu *accumulator* (A), *base* (B), *count* (C), dan *data* (D), dan masing-masing diberi akhiran X bila berukuran 16-bit: AX, BX, CX dan DX. Kemudian, masing-masing juga dapat dibagi menjadi dua, sebagai register 8-bit, dan diberi akhiran H dan L untuk menandakan *high* dan *low*: AH, AL, BH, BL, CH, CL, DH dan DL. Untuk pengolah Intel yang terbaru dengan menggunakan register 32-bit masing-masing diberi tambahan awalan E (*extended*): EAX, EBX, ECX dan EDX.

Register khusus terdiri dari register *cetakan* (*offset*) yang berfungsi menunjuk alamat memori per 16-bit atau 32-bit, register *penggal* (*segment*) yang berfungsi menunjuk alamat blok memori,

dan register bendera yang bertugas khusus sebagai rujukan kendali beberapa pernyataan. Semua ini akan dibicarakan secara lebih luas pada bab-bab berikutnya.

Komentar digunakan untuk menjelaskan maksud sekumpulan baris program atau satu baris program, setiap komentar didahului oleh tanda titik koma (;). Komentar tidak dirakit ke dalam bahasa mesin.

1.4 Merakit Program

Setelah penyandian dilaksanakan dengan baik, selanjutnya dilakukan perakitan program dengan menggunakan *perakit (assembler)*. Perakit mengambil berkas program – sandi sumber – dan menerjemahkan perintah-perintahnya ke dalam sandi mesin. Program ini membuka berkas-berkas program sumber dari tempat penyimpanan di mana disimpan berkas tersebut. Misalnya, dengan menggunakan Turbo Assembler, merakit dapat dilakukan dengan cara:

```
TASM [nama berkas program sumber]
```

Misalnya, untuk berkas 12.ASM dapat dirakit dengan mengetikkan TASM 12 [Enter].

Pada suguhan pertama melalui program sumber, perakit menentukan pemindahan butir-butir data dinamai, cetakan label, dan sebagainya, kemudian menempatkan informasi ini dalam suatu tabel simbol. Pada suguhan kedua melalui program sumber, perakit menghasilkan sandi binari untuk setiap perintah dan menyisipkan cetakan-cetakan, dan sebagainya, yang dihitung selama suguhan pertama, lihat senarai sandi berikut setelah dirakit dengan TASM// 12, di mana *l* mewakili *list*.

```
Turbo Assembler          Version 2.0          08/23/06 17:55:08
Page 1
12.ASM

    1 0000                                PengSandi SEGMENT
    2                                     ASSUME
CS: PengSandi
    3                                     ORG 100h
    4 0100 EB 29 90                        Mulai   : JMP Awal
    5 0103 50 72 6F 67 72 61              6D+   Pesan   DB
'Program pertama dengan bahasa rakitan',13,10,'$'
    6      20 70 65 72 74 61              6D+
    7      61 20 64 65 6E 67              61+
    8      6E 20 62 61 68 61              73+
```

```

    9          61 20 72 61 6B 69          74+
   10         61 6E 0D 0A 24

   11 012B   B4 09                          Awal   : MOV AH,09h
   12 012D   BA 0103r                         MOV DX,OFFSET
Pesan
   13 0130   CD 21                          INT 21h
   14 0132   CD 20                          INT 20h
   15 0134   PengSandi ENDS
   16                                     END Mulai

Turbo Assembler          Version 2.0          08/23/06 17:55:08
Page 2
Symbol Table

```

Symbol Name	Type	Value
??DATE	Text	"08/23/06"
??FILENAME	Text	"12"
??TIME	Text	"17:55:08"
??VERSION	Number	0200
@CPU	Text	0101H
@CURSEG	Text	PENG SANDI
@FILENAME	Text	12
@WORDSIZE	Text	2
AWAL	Near	PENG SANDI:012B
MULAI	Near	PENG SANDI:0100
PESAN	Byte	PENG SANDI:0103

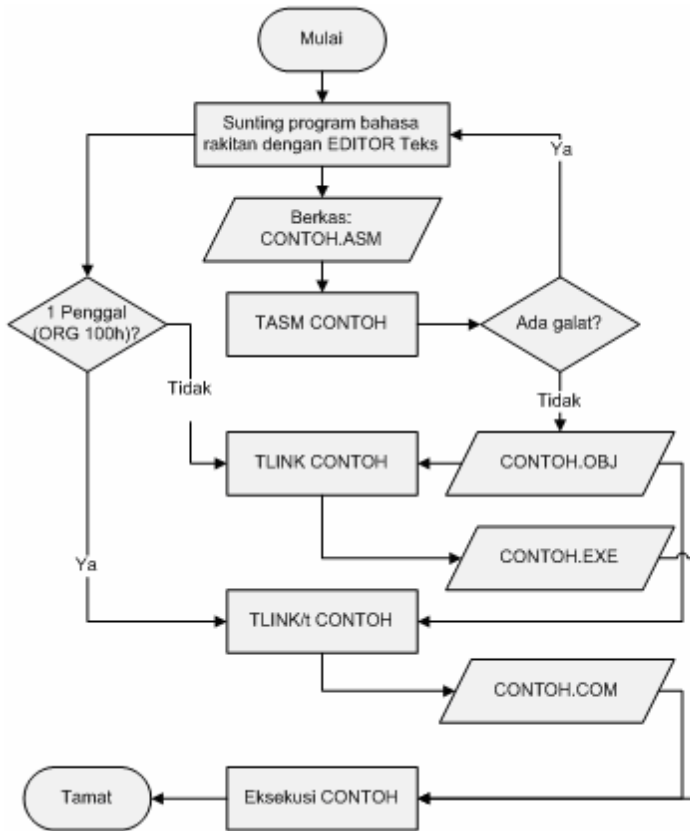
Groups & Segments	Bit	Size	Align	Combine	Class
PENG SANDI	16	0134	Para		none

Perakit menghasilkan dua berkas di tempat penyimpanan. Berkas pertama disebut *berkas tujuan*, diberi ekstensi .OBJ. Berkas ini mengandung sandi-sandi binari untuk perintah-perintah dan informasi tentang alamat-alamat perintah, karena sandi itu menjadi tujuan dari proses penerjemahan, yang berarti diberlakukan sebagai *sandi tujuan (object code)*. Penerjemah juga menghasilkan daftar program sumber dan menyatakan gangguan tentang aturan tata-bahasa dari bahasa pemrograman yang digunakan.

Setelah itu, dilakukan penjalinan dengan alat yang diberi nama *penjalın (linker)*, yaitu suatu program yang digunakan untuk menggabungkan beberapa berkas tujuan ke dalam satu berkas tujuan yang besar. Bila program yang ditulis cukup besar, biasanya melalui pembagian modul-modul.

Penjalın menghasilkan berkas jalinan yang mengandung sandi-sandi binari untuk semua modul yang digabungkan. Penjalın juga menghasilkan berkas pemetaan jalinan yang mengandung informasi alamat tentang berkas-berkas yang dijalin. Akan tetapi,

penjalın tidak menugaskan alamat-alamat mutlak ke program, kecuali hanya alamat-alamat relatif yang dimulai dengan nol. Bentuk program dikatakan dapat dilokasikan kembali sebab dapat ditempatkan di mana saja di dalam memori untuk dijalankan. Untuk SO, penjalın menghasilkan berkas berekstensi .EXE. Misalnya, penjalın Turbo Assembler dapat dijalankan dengan



Gambar 1.1 Cara merakit

TLINK [nama berkas tujuan]

Misalnya, untuk berkas 12.OBJ dapat dirakit dengan mengetikkan TLINK 12 [Enter].

Selain itu, pada beberapa perakıt memerlukan *penempat (locator)*, yang dulunya disediakan oleh SO. Penempat adalah program yang digunakan untuk menugaskan alamat khusus dari mana penggal sandi tujuan dimuat ke dalam memori. Namun, dengan menggunakan perakıt Turbo Assembler, penjalın yang disertakan

juga bertugas sebagai penempat. Jadi, untuk program sumber yang bangun strukturnya menggunakan satu penggal (64 Kbyte), dapat langsung dijalin dengan cara `TASM/t [nama berkas tujuan]`, `t` menunjukkan pengaktifan penempat, sehingga hasil jalinan berkas menjadi berekstensi `.COM`. Tata cara merakit program bahasa rakitan dengan Turbo Assembler, dapat dilihat pada diagram alir, Gambar 1.1.

1.5 Penutup

Teknik pemrograman atau rekayasa perangkat lunak dalam arti luas secara umum dapat dinyatakan sebagai urutan penyelesaian berikut: Pendefinisian masalah, penggambaran operasi-operasi program, penyandian dan mempersiapkan perintah, dan pengujian dan pencarian kesalahan. Jadi dengan proses berulang-ulang hingga program berakhir tanpa kesalahan dan dapat menyelesaikan persoalan yang dibebankan.

Langkah demi langkah penyelesaian tersebut memerlukan kelengkapan yang telah dibuat, mulai dari proses perancangan berdasarkan ide, menggambarkannya dalam suatu algoritma atau diagram alir, menerjemahkan ke dalam bahasa pemrograman yang disukai dan memeriksa kebenaran penyandiannya, serta mengetahui cara menjalankannya.

Soal.

1. Sebutkan dan jelaskan langkah-langkah pengembangan program?
2. Sebutkan tiga bagian program rakitan yang harus dinyatakan?
3. Sebutkan alat bantu pengembangan perangkat lunak yang anda ketahui melalui bahasa pemrograman ini?
4. Jelaskan hubungan perakit, penjalin dan penempat?
5. Senaraikan beberapa register serba guna?