11 PENYUSUNAN BAHASA RAKITAN TANPA DEBUG.COM

Cara lain untuk menuliskan program dengan bahasa rakitan adalah dengan menuliskan instruksi-instruksi bahasa rakitan melalui text editor dan kemudian diterjemahkan dengan assembler. Assembler merupakan program yang digunakan untuk menterjemahkan program aplikasi yang ditulis dengan bahasa rakitan (assembly language) atau bahasa pemrograman simbolik (simbolic programming language) menjadi bahasa mesin.

Instruksi program yang ditulis dengan menomonik akan diterjemahkan ke dalam bentuk bilangan biner bahasa mesin dengan menggunakan assembler. Program yang ditulis dengan bahasa simbolik tersebut disebut dengan source program (program sumber) dan hasil terjemahan ke dalam bahasa mesin disebut dengan object program (program obyek).

Berikut ini adalah bentuk umum struktur penulisan suatu segment.

[Nama Segment] SEGMENT

ASSUME CS:[Segment] DS:[Segment] SS:[Segment] ES:[Segment]

ORG 100h

```
[label]: ...
         ...
         ...
         Program ASM
         ...
         ...
         ...

[nama segment] ENDS
         END [label]
```

Untuk program .COM harus menambahkan ORG 100h pada baris ketiga dari struktur program di atas. Hal ini dilakukan karena harus menyisihkan ruang sebesar 100h untuk PSP (Program Segment Prefix). Karena program .COM hanya terdiri dari satu segment, maka bentuk struktur program dari program .COM adalah sama seperti di atas. Sedangkan untuk program .EXE harus dapat membuat beberapa segment dan tidak perlu menyediakan baris ketiga dari struktur data di atas.

Berikut ini adalah contoh program dan penjelasannya.

```
COMMENT *
    Cetak tulisan 0123456789 di layar
    dengan nama executable program Contoh1.COM *

Code_seg SEGMENT
            ASSUME CS:Code_seg DS:Code_seg SS:Code_seg ES:Code_seg
            ORG 100h

Mulai:  MOV CX,0Ah          ; jumlah loop
            MOV DL,30h      ; ASCII huruf 0

Label:  MOV AH,02h
        INT 21h
        INC DL
        LOOP Label
        INT 20h

Code_seg ENDS
            END     Mulai
```

Penjelasan:

1. COMMENT merupakan suatu pseudo-op yang dapat digunakan untuk memberi keterangan atau komentar dari program.

   Format dari pseudo-op COMMENT adalah sebagai berikut

   COMMENT pembatas awal keterangan pembatas akhir

2. SEGMENT merupakan pseudo-op yang menunjukkan segment main memory letak dari instruksi dan data di program assembly. SEGMENT didahului dengan label yang menunjukkan nama dari segmentnya.

   Code_seg SEGMENT
   Code_seg ENDS

   Pada contoh program, nama dari segment adalah Code_seg (tidak harus Code_seg tetapi bisa nama yang lainnya, karena merupakan suatu label).

   Untuk program .EXE harus dapat membuat beberapa segment dan tidak perlu menyediakan baris ini dari struktur data di atas.

3. ASSUME merupakan pseudo-op yang menunjukkan kepada assembler letak dari segment untuk *segment segment register.*

   Karena semua penunjuk ke Code_seg sebenarnya baris ini dapat kita ganti dengan kata kata yang lebih singkat, yaitu:

   ASSUME CS:Code_seg

   Untuk program .EXE harus dapat menunjukkan ke mana arahnya penunjuk penunjuk itu, contoh:

   ASSUME CS:Code_seg DS:Data_seg SS:Stack_seg ES: 0B690H

4. ORG merupakan pseudo-op yang menunjukkan letak awal dari alamat memori dalam suatu segment.

   Bentuk instruksi : ORG 100H

   artinya instruksi ini akan diletakkan mulai dari alamat memori 100 heksadesimal (=256 Des) pada code segment. Hal ini diperlukan untuk executable program yang mempunyai extension.COM, yang umumnya diletakkan pada lamat memori 100H di-segment.

Executable program dengan yang berextension.COM lebih kecil dan dapat diambil lebih cepat dibandingkan dengan extension.EXE, hanya kelemahannya dapat menampung maksimal 64 KB.

5. MULAI merupakan label yang mutlak diperlukan untuk menunjukkan awal dari instruksi-instruksi yang akan diproses.

MOV merupakan op-code yang digunakan untuk memindahkan suatu nilai ke suatu lokasi.

Format dari op-code MOV adalah:

MOV tujuan, sumber

Bentuk instruksi pada contoh adalah:

MOV AH,0A ; fungsi DOS

Hal ini mempunyai arti bahwa register AH diisi dengan nilai data 10 desimal. Mengapa register AH harus diisi dengan nilai tersebut ? Tentunya hal ini mem-punyai maksud yang tertentu. Untuk menampilkan suatu bentuk karakter string di layar terminal, assembler tidak menyediakan suatu intruksi yang khusus, tetapi dapat dilakukan dengan menggunakan fasilitas dari DOS. Fasilitas dari DOS dapat diaktipkan dengan menggunakan op - code INT (interrupt). Salah satu interrupt DOS yang penting adalah dengan memberikan nilai 21 hexadesimal, yang menyediakan beberapa fungsi operasi yang sangat berguna .Untuk menggunakan fungsi ini,suatu nilai fungsi tertentu harus diletakkan di register AH.

Sebelum dilakukan interupt terhadap DOS, maka register DX (data register) diisi terlebih dahulu.

6. Label

Struktur dapat dituliskan sebagai berikut:

[label:] [perintah] [;Komentar]

Syarat untuk label yang terdapat dalam suatu baris adalah:

a. Harus unik.
b. maksimal 31 karakter
c. karakter yang digunakan adalah: alphanumerik, tanda tanya, tanda satuan (@), titik, tanda dolar.
d. bukan merupakan perintah assembler.
e. label hanya dapat didefinisikan satu kali dalam program

Label yang terdapat pada struktur kalimat program dapat digunakan sesuai keperluan. Sebagai contoh perhatikan program di atas,pada baris kelima terlihat bahwa perintah tersebut menggunakan label, namun pada baris dibawahnya tidak lagi menggunakan label.

Komentar yang dimaksudkan dalam struktur kalimat program adalah suatu yang tidak akan diproses oleh compiler dan akan dibiarkan sebagai keterangan pada file ASM. Setiap komentar selalu dimulai dengan tanda ';'. Sebagai contoh baris-baris pada program di atas menggunakan komentar. Sebenarnya tidak setiap baris harus diberi komentar. Pemberian komentar sebenarnya terserah saudara dan bila perlu dapat tidak memberikan komentar sama sekali pada suatu program.

7. ENDS

Instruksi : Code_seg ENDS

Instruksi ini merupakan perintah tanda batas dari suatu segment.

8. END

Instruksi : END Mulai

Instruksi ini merupakan tanda berakhirnya suatu program. Untuk mengakhiri program ini Saudara harus membubuhkan label pertama tanda memulai program, dalam hal ini adalah label Mulai.

Catatan:

Pada label dapat diberikan instruksi berikut:

Instruksi : MOV DX, OFFSET baca

Instruksi ini menunjukkan bahwa register DX agar diisi dengan nilai yang ada pada label yang ditunjukkan oleh operator OFFSET, yaitu nilai dimana label baca. Pada baris dimana label baca, terdapat psudo-op DB (Define Byte) yang digunakan untuk men-definisikan berupa string, maka diakhiri dengan tanda dolar ($), seperti berikut:

```
baca DB 'GUNADARMA$'
```

Berikut ini diberikan contoh yang menggunakan instruksi DB .

```
COMMENT *
    tampilkan tulisan GUNADARMA pada monitor
    dengan nama executable program Contoh2.COM *
```

```
Kode    SEGMENT
        ASSUME CS:kode, DS:Kode
        ORG 100H

Mulai:  MOV AH,09               ; fungsi DOS
        MOV DX,OFFSET baca      ; isi baca di DX
        INT 21H                 ; cetak dimonitor
        INT 20H                 ; selesai

bacaDB 'GUNADARMA$'

kode    ENDS
        END Mulai
```