

PRAKTIKUM MIKROPROSESOR DAN BAHASA RAKITAN

OPERASI PADA LAYAR

TUJUAN

1. Mahasiswa memahami dan mengerti tentang berbagai operasi pada layar serta penerapannya dalam bahasa mesin

DASAR TEORI

Layar merupakan sarana penampil informasi yang amat menarik yang kaya akan berbagai fitur. Layar merupakan sebuah kumpulan memori khusus yang nantinya akan ditampilkan ke pengguna. Memori khusus ini akan dibaca dan ditampilkan secara konstan kurang lebih 70 kali per detik atau sering disebut sebagai 70 frame per seconds. Kecepatan ini sering disebut sebagai refresh rate. Letak dari memori layar ini yaitu pada alamat B000H:0000 untuk layar monokrom, dan pada alamat B800H:0000 untuk layar berwarna. Pengertian dasar tentang memori layar ini amat penting untuk dipahami agar nantinya dapat mempermudah dalam pembuatan programnya.

Untuk menampilkan teks dengan atributnya pada layar maka dibutuhkan 2 byte data yaitu B800H:0000 – B800H:0001 untuk menampilkan data teks dengan atributnya untuk baris 0 kolom 0 layar. Untuk baris 0 kolom 1 maka datanya yaitu pada alamat B800H:0002 – B800H:0003. Hal ini akan diulang terus sampai seluruh layar terisi pikselnya. Pada mode defaultnya maka layar terbagi menjadi 80 kolom dan 25 baris, jadi total memori yang dibutuhkan yaitu sebesar 4kB. Dengan pengalamatan memori yang berurutan seperti ini, maka alamat memori untuk menampilkan data ke layar dengan atribut tertentu dapat dirumuskan sebagai berikut :

$$\text{Offset karakter} = (\text{baris} * 160) + (\text{kolom} * 2)$$

$$\text{Offset atribut} = (\text{baris} * 160) + (\text{kolom} * 2) + 1$$

BIOS menyediakan suatu fungsi yang dapat digunakan untuk menggulung layar dengan batasan yang kita tentukan. Adapun aturan pemakaian dari interupsi ini adalah:

INPUT:

AH = Diisi dengan 6 untuk menggulung layar keatas, untuk menggulung layar kebawah diisi dengan 7.

AL = Banyaknya pergeseran yang akan dilakukan. Jika diisi dengan

nol, maka seluruh isi window akan dihapus.

CH = Posisi baris kiri atas window

CL = Posisi kolom kiri atas window

DH = Posisi baris kanan bawah window

DL = Posisi kolom kanan bawah window

BH = Atribut yang akan mengisi hasil penggulungan window

Untuk memindahkan posisi kursor, sama halnya dengan perintah GOTOXY pada pascal, bisa anda gunakan interupsi dari BIOS. Interupsi yang digunakan adalah interupsi 10h dengan aturan pemakaian:

INPUT:

AH = 2

DH = Posisi Baris(00-24)

DL = Posisi Kolom(00-79)

BH = Halaman layar(0=default)

Sama halnya dengan fungsi WhereX dan WhereY dalam pascal, didalam assembler anda juga bisa mengetahui posisi dari kursor. Untuk itu telah tersedia interupsi 10h dari BIOS dengan aturan pemakaian:

INPUT: OUTPUT:

AH = 03 DH = Posisi Baris

BH = Halaman Layar(0=default) DL = Posisi Kolom

Telah kita bahas bahwa pada normalnya satu layar akan menggunakan 4000 byte memory. Tetapi memory yang disediakan untuk layar ini sebenarnya malah beberapa kali lipat lebih banyak dari 4000 byte, karenanya terciptalah apa yang dinamakan 'paging' atau halaman tampilan layar. Banyaknya halaman tampilan ini sangat bervariasi karena tergantung jumlah memory yang tersedia dan jumlah memory yang digunakan oleh satu halaman tampilan. Untuk mengaksesnya maka dapat dipergunakan rumusan berikut ini :

Offset Karakter= (Baris * 160)+(Kolom * 2) + (Halaman*1000h)

Offset Atribut = (Baris * 160)+(Kolom * 2)+1+(Halaman*1000h)

Secara default halaman tampilan yang digunakan adalah halaman tampilan ke 0, yang beralamat awal pada B800:0000h. Untuk merubah halaman tampilan yang aktif ini bisa anda gunakan servis 5 dari interupsi 10h. Adapun aturan pemakaian servis ini adalah:

INPUT:

AH = 5

AL = Nomor halaman tampilan yang akan diaktifkan

Bila monitor anda adalah monitor EGA keatas, maka bentuk karakter bisa diubah dengan mengubah isi dari tabel yang menyusun karakter-karakter ini. Untuk itu BIOS telah menyediakan interupsi 10h, service 11h, subservis 00 untuk keperluan ini. Adapun aturan dari pemakaiannya adalah:

INPUT:

AH = 11h

AL = 00h

CX = Jumlah bentuk karakter yang akan diganti

DX = Kode ASCII karakter awal yang akan diganti

BL = Nomor tabel karakter yang diubah

BH = Jumlah byte per karakter

ES:BP = Alamat buffer pola karakter

PROSEDUR PERCOBAAN

A. Menampilkan karakter dengan operasi memori layar

1. Tuliskan program *assembly* berikut menggunakan TASM, lalu *lingking* menjadi COM :

```
Tulis_Kar MACRO X,Y,Kar,Attr
MOV AX,0B800h
MOV ES,AX ; ES Menunjuk pada segment layar
MOV AH,Y
MOV AL,160
MUL AH ; Hitung offset baris
MOV BX,AX ; Simpan hasilnya pada BX
MOV AH,X
MOV AL,2
MUL AH ; Hitung offset kolom
ADD BX,AX ; Tambahkan hasilnya pada BX
MOV AL,Kar ; AL=karakter yang akan ditampilkan
MOV AH,Attr ; AH=Atribut yang akan ditampilkan
MOV ES:[BX],AL ; Tampilkan Karakter dan atributnya
MOV ES:[BX+1],AH ; pada posisi kolom X dan baris Y
ENDM
;/=====\;
; Program : LAYAR1.ASM ;
; ;
; Fungsi : Menampilkan karakter dan atributnya ;
; dengan menuliskannya langsung pada ;
; memory layar ;
```

```

; \===== /;
.MODEL SMALL
.CODE
ORG 100h
Proses :
Tulis_Kar 40 12 'S' 95 ; Tulis karakter 'S' dengan
; no atribut 95 pada posisi
INT 20h ; kolom 40 dan baris 12
END Proses

```

2. Jalankan programnya dan lihat hasilnya.

B. Program Rontok

1. Tuliskan program *assembly* berikut menggunakan TASM, lalu *lingking* menjadi COM :

```

Delay MACRO
PUSH CX ; Macro ini digunakan untuk
XOR CX,CX ; menunda program, dan
Loop1:
LOOP Loop1 ; hanya melakukan looping
POP CX
ENDM
Geser MACRO PosY
PUSH AX
PUSH BX
PUSH CX ; Simpan semua register yang digunakan
XOR CX,CX
MOV AL,26
SUB AL,PosY
MOV CL,AL ; CX=banyaknya pergeseran kebawah
Loop2:
MOV AL,BYTE PTR ES:[BX] ; AL=Karakter pada layar
MOV BYTE PTR ES:[BX+160],AL ; Geser ke bawah
Hilang:
MOV BYTE PTR ES:[BX], ' ' ; Hapus karakter
; sebelumnya
Delay ; delay, supaya bisa
; terlihat
ADD BX,160 ; Menuju baris selanjutnya
LOOP Loop2 ; Ulangi ke Loop2
POP CX
POP BX
POP AX ; Kembalikan semua register yang digunakan
ENDM

;/===== \;
; Program : RONTOK.ASM ;
; Fungsi : Membersihkan layar dengan cara ;
; merontokkan hurufnya satu persatu;
; ;
; \===== /
.MODEL SMALL
.CODE
ORG 100h

```

```

TData : JMP Proses
PosY DB ?
Proses:
MOV AX,0B800h
MOV ES,AX ; ES mencatat segment layar
MOV BX,3998 ; Posisi karakter 80,25
MOV CX,25 ; Banyaknya pengulangan baris
UlangY :
MOV PosY,CL ; PosY mencatat posisi baris
PUSH CX ; CX mencatat posisi Y
MOV CX,80 ; Banyaknya pengulangan Kolom
UlangX :
CMP BYTE PTR ES:[BX],33 ; Apakah ada karakter
; pada layar ?
JB Tdk ; Lompat ke Tdk, jika tidak ada
Geser PosY ; Geser karakter tersebut ke bawah
Tdk :
SUB BX,2 ; BX menunjuk karakter selanjutnya
LOOP UlangX ; Proses 80 kali untuk kolom
POP CX ; Ambil posisi Y
LOOP UlangY ; Ulangi dan ganti baris ke atas
EXIT:
INT 20h
END TData

```

2. Jalankan programnya dan amati yang terjadi.

C. Menggulung layar ke atas

1. Tuliskan program *assembly* berikut menggunakan TASM , lalu *linking* menjadi COM :

```

DELAY MACRO ; Macro untuk menunda program
LOCAL Ulang
PUSH CX
XOR CX,CX
Ulang:
LOOP Ulang
POP CX
ENDM
Scrool MACRO X1,Y1,X2,Y2,Arah
PUSH CX
MOV AH,Arah ; Servis Gulung keatas atau kebawah
MOV AL,1 ; Jumlah Baris
MOV CL,X1 ; Kolom kiri atas
MOV CH,Y1 ; Baris kiri Atas
MOV DL,X2 ; Kolom kanan bawah
MOV DH,Y2 ; Baris kanan bawah
MOV BH,01000111b ; Atribut hasil penggulungan
INT 10h
POP CX
ENDM
;/=====\;
; Program : SCROOL.ASM ;
; Fungsi : Menggulung layar ;
;\=====;/;

```

```

.MODEL SMALL
.CODE
ORG 100h
TData : JMP Proses
G_Atas EQU 6 ; Servis untuk menggulung ke atas
G_Bawah EQU 7 ; Servis untuk menggulung ke bawah
Proses:
MOV CX,7
Ulang:
Scroll 20 7 60 14 G_Bawah
delay
LOOP Ulang
INT 20h
END TData

```

2. Jalankan programnya dan amati yang terjadi.

D. Menu sorot

1. Tuliskan program *assembly* berikut menggunakan TASM , lalu *linking* menjadi COM :

```

Cls MACRO ; Macro untuk menghapus layar
MOV AX,0600h
XOR CX,CX
MOV DX,184Fh
MOV BH,10 ; Atribut Hijau diatas hitam
INT 10h
ENDM
GotoXY MACRO X,Y ; Macro untuk memindahkan kursor
MOV AH,02
XOR BX,BX
MOV DH,Y
MOV DL,X
INT 10h
ENDM
SimpanL MACRO ; Macro untuk menyimpan seluruh
LOCAL Ulang ; isi layar monitor
MOV AX,0B800h
MOV ES,AX
MOV CX,4000
XOR BX,BX
Ulang:
MOV AL,ES:[BX]
MOV Layar[BX],AL
INC BX
LOOP Ulang
ENDM
BalikL MACRO ; Macro untuk mengembalikan semua
LOCAL Ulang ; isi layar yang telah disimpan
MOV CX,4000
XOR BX,BX
Ulang:
MOV AL,Layar[BX]
MOV ES:[BX],AL
INC BX

```

```

LOOP Ulang
ENDM
Sorot MACRO X,Y ; Macro untuk membuat sorotan
LOCAL Ulang ; pada menu
MOV BL,Y
MOV AL,160
MUL BL
MOV BX,AX
MOV AL,X
MOV AH,2
MUL AH
ADD BX,AX
INC BX ; Alamat warna pada posisi X,Y
MOV CX,25 ; Panjangnya sorotan
Ulang:
MOV BYTE PTR ES:[BX],4Fh ; Atribut sorotan
; putih diatas merah
ADD BX,2
LOOP Ulang
ENDM
Readkey MACRO ; Macro untuk membaca masukan dari
MOV AH,00 ; keyboard.
INT 16h ; hasilnya AH=Extended, AL=ASCII
ENDM
MenuL MACRO String ; Macro untuk mencetak menu
MOV AH,09
LEA DX,String
INT 21h
ENDM
; /=====\;
; Program : SOROT.ASM ;
; Fungsi : Membuat menu sorot untuk ;
; digunakan program ;
; \=====;/
.MODEL SMALL
.CODE
ORG 100h
TData: JMP Proses
Layar DB 4000 DUP (?)
Menu DB 9,9,'+=====+',13,10
DB 9,9,'| >>>> MENU SOROT <<<< |',13,10
DB 9,9,'+=====+',13,10
DB 9,9,'| |',13,10
DB 9,9,'| 1. Pilihan pertama |',13,10
DB 9,9,'| 2. Pilihan Kedua |',13,10
DB 9,9,'| 3. Pilihan Ketiga |',13,10
DB 9,9,'| 4. Pilihan Keempat |',13,10
DB 9,9,'| |',13,10
DB 9,9,'+=====+'$'
PosX DB 22 ; Posisi kolom mula-mula
PosY DB 12 ; Posisi baris mula-mula
Panah_Atas EQU 72 ; Kode tombol panah atas
Panah_Bawah EQU 80 ; Kode tombol panah bawah
TEnter EQU 0Dh ; Kode tombol Enter
Proses :
Cls ; Hapus layar
GotoXY 0 8 ; kursor = 0,8

```

```

MenuL Menu ; Gambar menu
SimpanL ; Simpan isi layar
Ulang :
BalikL ; Tampilkan isi layar yang
; disimpan
Sorot PosX,PosY ; Sorot posisi X,Y
Masukan:
Readkey ; Baca masukan dari keyboard
CMP AH,Panah_Bawah ; Panah bawah yang ditekan ?
JE Bawah ; Ya! lompat bawah
CMP AH,Panah_Atas ; Panah atas yang ditekan ?
JE CekY ; Ya, lompat CekY
CMP AL,TEnter ; Tombol enter yang ditekan ?
JNE Masukan ; Bukan, lompat ke ulangi
JMP Selesai ; Ya, lompat ke selesai
CekY :
CMP PosY,12 ; Apakah sorotan paling atas ?
JE MaxY ; Ya! lompat ke MaxY
DEC PosY ; Sorotkan ke atas
JMP Ulang ; Lompat ke ulang
MaxY :
MOV PosY,15 ; PosY=Sorotan paling bawah
JMP Ulang ; lompat ke ulang
Bawah :
CMP PosY,15 ; apakah sorotan paling bawah ?
JE Noly ; Ya! lompat ke Noly
INC PosY ; Sorotkan ke bawah
JMP Ulang ; Lompat ke ulang
Noly :
MOV PosY,12 ; Sorotan paling atas
JMP Ulang ; Lompat ke ulang
Selesai:
INT 20h
END TData

```

2. Jalankan programnya dan amati yang terjadi.

E. Merubah halaman layar

1. Tuliskan program berikut dan simpan sebagai macku.mcr di dalam folder yang sama dengan

TASM.EXE :

```

Delay MACRO Rep ; Macro ini untuk menunda program
LOCAL Ulang
PUSH CX
MOV DX,Rep
SUB CX,CX
Ulang:
LOOP Ulang
DEC DX
CMP DX,0
JNZ Ulang
POP CX
ENDM
Ak_Page MACRO No ; Macro ini digunakan untuk

```



```

MOV AH,5 ; mengaktifkan halaman layar
MOV AL,No
INT 10h
ENDM

; /===== \;
; Program : PAGE.ASM ;
; Fungsi : Untuk mengaktifkan halaman;
; layar tertentu ;
; \===== /;
.MODEL SMALL
.CODE
ORG 100h
Tdata : JMP Proses
Kal0 DB 'INI ADALAH HALAMAN TAMPILAN KE 2 ',13,10
DB ' DENGAN ALAMAT AWAL B800:1000h !!! $'
Proses:
Ak_Page 2 ; Aktifkan halaman layar yang ke 2
MOV AH,09 ;
LEA DX,Kal0 ; Tulis kalimat pada halaman ke 2
INT 21h ;
MOV CX,3 ; Banyaknya pengulangan
Ulang:
Ak_Page 2 ; Aktifkan halaman ke 2
Delay 100
Ak_Page 0 ; Aktifkan halaman ke 0
Delay 100
LOOP Ulang
INT 20h
END Tdata

```

2. Jalankan programnya dan amati yang terjadi.