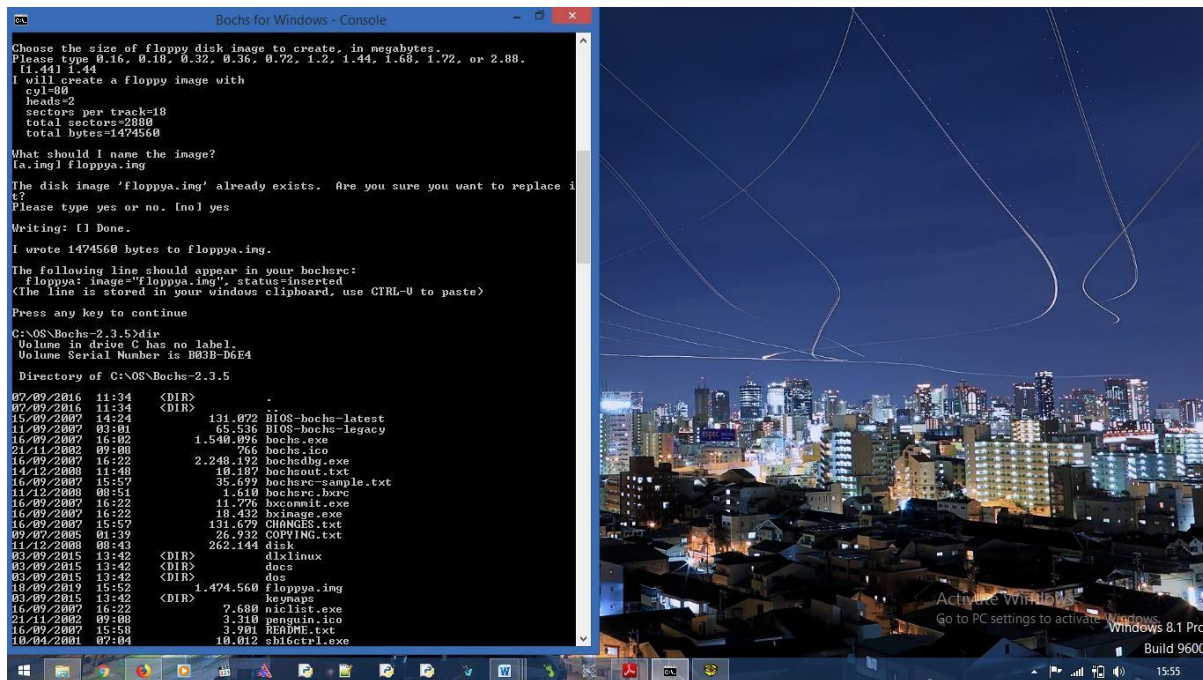
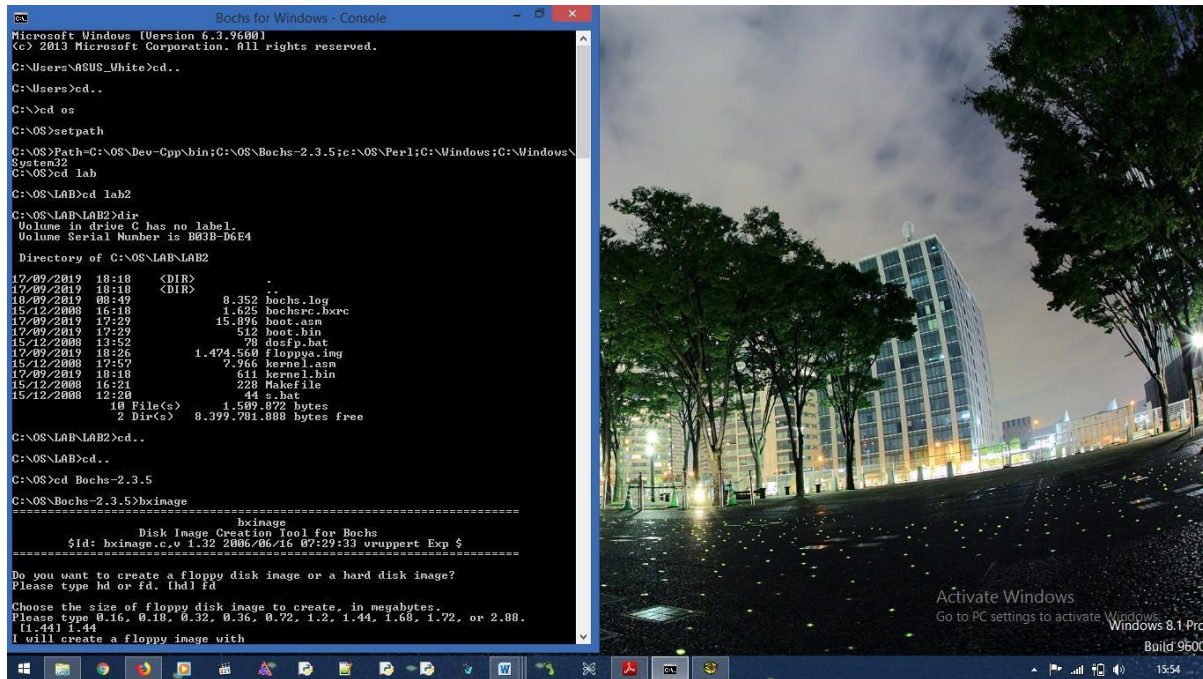


Nama : Hanif Firdaus Illahi
NIM : L200180021




```
bootasm - Notepad
File Edit Format View Help
; *****
; LAB-2 : boot-strap loader - real mode
; digunakan untuk memindahkan file OS/kernel
; dari floppy disk ke dalam RAM (memori kerja PC).
; *****

;Mengatur mode kerja 16 bit (real-mode)
[BITS 16]

;Menentukan lokasi awal dari program
[ORG 0x0000]

;=====
;(1) BLOK STARTER
;   loncat ke label START
;=====
jmp     START

;=====
;(2) BLOK ID FAT12
;   Keterangan format FAT12 pada floppy disk
;=====

OEM_ID      db "MY  -OS"
BytesPerSector dw 0x0200
SectorsPerCluster db 0x01
ReservedSectors dw 0x0001
TotalFATS     db 0x02
MaxRootEntries dw 0x00E0
TotalSectorsSmall dw 0x0B40
MediaDescriptor db 0xF0
SectorsPerFAT dw 0x0009
SectorsPerTrack dw 0x0012
NumHeads      dw 0x0002
HiddenSectors dd 0x00000000
TotalSectorsLarge dd 0x00000000
```

```
bootasm - Notepad
File Edit Format View Help

TotalSectorsLarge dd 0x00000000
DriveNumber       db 0x00
Flags             db 0x00
Signature         db 0x29
VolumeID          dd 0xFFFFFFFF
VolumeLabel       db "QUASI  BOOT"
SystemID          db "FAT12  "

;=====
;(3) Blok BOOT CODE
;=====

START:
; Mengatur lokasi kode program pada alamat 7C00:0000, dan mengatur REGISTER SEGMENT
cli                                ; matikan aktifitas interupsi
mov     ax, 0x07C0
mov     ds, ax
mov     es, ax
mov     fs, ax
mov     gs, ax

; Mengatur lokasi stack
mov     ax, 0x0000
mov     ss, ax
mov     sp, 0xFFFF ; sp bergerak dari alamat atas ke bawah
sti                                ; aktifkan aktifitas interupsi

; Menampilkan text di layar
mov     si, msgLoading ; mengambil lokasi text yang di simpan dalam 'msgloading'
call    DisplayMessage

LOAD_ROOT:
; menghitung ukuran 'root directory' dan menyimpannya dalam register 'cx'
xor     cx, cx
xor     dx, dx
mov     ax, 0x0020 ; Ukuran satu nama direktori sepanjang 32 byte
```

```

bootasm - Notepad
File Edit Format View Help

mov ax, 0x0020 ; Ukuran satu nama direktori sepanjang 32 byte
mul WORD [MaxRootEntries] ; Total lokasi direktori 32 x 224 (heksa 0x00E0)=7168
div WORD [BytesPerSector] ; lokasi sektor yang digunakan untuk menyimpan direktori 7168/512 (0x0200) = 14
xchg ax, cx ; Ukuran 'root direktori' = 14 sektor

; menghitung jumlah sektor yang digunakan untuk menyimpan FAT
; untuk mencari lokasi awal sektor ROOT DIREKTORI di simpan di register 'ax'
; hasil disimpan pada variabel 'datasector'
mov al, BYTE [TotalFATs] ; jumlah FAT (2 copy)
mul WORD [SectorsPerFAT] ; dikalikan dengan jumlah sektor yang digunakan FAT (9 sektor/FAT)
add ax, WORD [ReservedSectors] ; tambah cadangan sektor (1 sektor) simpan di 'ax'
mov WORD [datasector], ax ; pindahkan hasilnya ke variabel 'datasector' (=19 sektor)
add WORD [datasector], cx ; tambah dengan ukuran 'root direktori' (14),
; jadi total 33 sektor untuk menyimpan 2 FAT (Field Allocation Table)
; dan data 'ROOT DIREKTORI'

; Memindahkan isi root direktori ke memori RAM pada alamat 7C00:0200
;
mov bx, 0x0200 ; lokasi awal memori untuk menampung pembacaan sektor
call ReadSectors

; mencari nama file OS
;
mov cx, WORD [MaxRootEntries] ; baca semua file dalam root direktori
mov di, 0x0200 ; lokasi awal berisi copy ROOT DIREKTORI

.LOOP:
push cx
mov cx, 0x000B ; ukuran nama FILE dan DIREKTORI sebanyak 11 karakter
mov si, ImageName ; Mengambil nama FILE yang di cari
; nama FILE tersimpan dalam variabel ImageName
; file diberi nama 'MYKERNELBIN'

push di
rep cmpsb ; Bandingkan dengan daftar file root direktori

```

```

bootasm - Notepad
File Edit Format View Help

rep cmpsb ; Bandingkan dengan daftar file root direktori
pop di
je LOAD_FAT ; jika file sudah ditemukan loncat ke label LOAD_FAT
pop cx
add di, 0x0020 ; Nama file berikutnya
loop .LOOP ; periksa lagi
jmp FAILURE ; Jika nama file tidak ada loncat ke label FAILURE

LOAD_FAT:
;
; Simpan nomor awal cluster tempat menyimpan data file 'MYKERNELBIN'
mov si, msgCRLF
call DisplayMessage
mov dx, WORD [di + 0x001A]
mov WORD [cluster], dx ; Cluster pertama file 'MYKERNELBIN',
; byte ke 0x1A (26) dari 32 byte, dibelakang nama file
; register 'di' menunjuk lokasi byte pertama dari namafile

; Menghitung jumlah sektor yang digunakan untuk menyimpan FAT
; hasilnya disimpan pada register 'cx'
xor ax, ax
mov al, BYTE [TotalFATs] ; Jumlah FAT, pada FAT12 ada 2 copy FAT
mul WORD [SectorsPerFAT] ; dikalikan dengan jumlah sektor/FAT (9)
mov cx, ax

; Mencari nomor awal sektor lokasi FAT
; hasil disimpan di reg 'ax'
mov ax, WORD [ReservedSectors] ; sesuaikan dengan jumlah sektor cadangan (1 sektor)

; baca isi FAT pindahkan ke lokasi memory (7C00:0200)
mov bx, 0x0200 ; pindahkan FAT di atas lokasi bootcode
call ReadSectors

; baca isi file 'KERNEL BIN' pindahkan ke memori dengan lokasi (0100:0000)
mov si, msgCRLF
call DisplayMessage ; menampilkan text di layar
mov ax, 0x0100 ; atur segmen kode target CS = 0x0100

```

```

bootasm - Notepad
File Edit Format View Help

    mov ax, 0x0100                ; atur segmen kode target CS = 0x0100
    mov es, ax
    mov bx, 0x0000                ; atur target IP= 0x0000
    push bx

LOAD_IMAGE:
    mov ax, WORD [cluster]        ; Nomor cluster awal 'KERNEL BIN'
    pop bx                        ; Lokasi awal memori untuk menyimpan 'KERNEL BIN'
    call ClusterLBA                ; konversi dagta cluster to LBA (nomor sektor)
    ; hasil di simpan di 'ax' (nomor sektor yang harus di baca)

    xor cx, cx
    mov cl, BYTE [SectorsPerCluster] ; Jumlah sektor yang di baca
    call ReadSectors
    push bx

; Baca cluster berikutnya
    mov ax, WORD [cluster]        ; identifikasi posisi cluster saat ini
    mov cx, ax                    ; copy cluster saat ini ke 'cx'
    mov dx, ax                    ; copy cluster saat ini ke 'dx'
    shr dx, 0x0001                ; bagi dengan 2 (posisi bit di geser ke kanan 1 digit)

; nomor sekarang di tambah (1/2 nomor cluster)
    add cx, dx                    ; (3/2) nilai cluster saat ini
    mov bx, 0x0200                ; lokasi FAT di memory
    add bx, cx                    ; index FAT
    mov dx, WORD [bx]             ; baca dua byte dari FAT
    test ax, 0x0001
    jnz .ODD_CLUSTER

.EVEN_CLUSTER:
    and dx, 0000111111111111b    ; jika angkanya genap ambil 12 bit terbawah
    jmp .DONE
.ODD_CLUSTER:
    shr dx, 0x0004                ; jika angkanya ganjil ambil 12 bit teratas
.DONE:
    mov WORD [cluster], dx        ; simpan nomor cluster yang baru
    cmp dx, 0x0FF0                ; Periksa sudah mencapai bagian akhir file
    jb LOAD_IMAGE

```

```

bootasm - Notepad
File Edit Format View Help

    jb LOAD_IMAGE
.DONE:
    mov si, msgCRLF
    call DisplayMessage
    push WORD 0x0100                ; Loncat ke alamat 0100:0000
    push WORD 0x0000
    retf

FAILURE:
    mov si, msgFailure
    call DisplayMessage
    mov ah, 0x00
    int 0x16                        ; Tunggu penekanan tombol keyboard
    int 0x19                        ; reboot

; *****
; PROCEDURE DisplayMessage
; Menampilkan string yang tersimpan pada alamat yang ditunjuk oleh 'ds:si'
; menggunakan program pada BIOS
; *****
DisplayMessage:
    lodsb                          ; Ambil karakter berikutnya
    or al, al                      ; periksa bagian akhir data (apakah karakter NUL?)
    jz .DONE
    mov ah, 0x0E                   ; BIOS teletype
    mov bh, 0x00                   ; display page 0
    mov bl, 0x07                   ; text attribute
    int 0x10                       ; panggil BIOS
    jmp DisplayMessage
.DONE:
    ret

; *****
; PROSEDURE ReadSectors
; Membaca sektor sebanyak angka yang tersimpan pada register 'cx'
; mulai dari nomor sektor, sesuai angka yang tersimpan pada register 'ax'
; dipindahkan ke lokasi memori yg ditunjuk oleh kombinasi register 'es:bx'
; *****

```



```

bootasm - Notepad
File Edit Format View Help
;*****
ReadSectors:                ; nama prosedur
.MAIN
    mov     di, 0x0005        ; mencoba membaca sektor sebanyak 5 kali
                                ; untuk memastikan adanya Error pada DISK

.SECTORLOOP
    push    ax
    push    bx
    push    cx
    call    LBACHS
    mov     ah, 0x02          ; Perintah BIOS untuk membaca sektor
    mov     al, 0x01          ; jumlah sektor yang di baca (1 sektor)
    mov     ch, BYTE [absoluteTrack] ; Nomor track, absoluteTrack=0
    mov     cl, BYTE [absoluteSector] ; Nomor sektor, absoluteSector = 0
    mov     dh, BYTE [absoluteHead]   ; Nomor head, absoluteHead = 0, head ada 2 dengan nomor 0 dan 1
    mov     dl, BYTE [DriveNumber]    ; Nomor drive, 'DriveNumber=0
    int     0x13              ; memanggil BIOS untuk membaca sektor

    jnc     .SUCCESS          ; Tes hasil pembacaan sektor
                                ; jika C=0 (sukses) loncat ke label '.SUCCESS'
    xor     ax, ax            ; pilih fungsi BIOS untuk mereset disk
    int     0x13              ; panggil BIOS
    dec     di                ; kurangi angka jumlah pengujian baca sektor
    pop     cx
    pop     bx
    pop     ax
    jnz     .SECTORLOOP       ; jika ada error sebelumnya coba baca lagi
    int     0x18

.SUCCESS
    mov     si, msgProgress    ; menampilkan status proses baca sektor
    call    DisplayMessage
    pop     cx
    pop     bx
    pop     ax
    add     bx, WORD [BytesPerSector] ; lokasi memori selanjutnya
    inc     ax                ; nomor sektor berikutnya
    loop    .MAIN             ; baca sektor berikutnya
;*****

```

```

bootasm - Notepad
File Edit Format View Help
loop .MAIN                    ; baca sektor berikutnya
ret

;*****
; PROSEDURE ClusterLBA
; Merubah data nomor cluster menjadi skema alamat LBA
; dengan rumus sebagai berikut
; LBA = (cluster - 2) * sectors per cluster
; nomor cluster di simpan pada register 'ax'
;*****
ClusterLBA:
    sub     ax, 0x0002        ; nomor cluster - 2, hasil disimpan di 'ax'
    xor     cx, cx
    mov     cl, BYTE [SectorsPerCluster] ; konversi dari byte to word
    mul     cx
    add     ax, WORD [datasector] ; Nomor sektor pertama lokasi data file 'MYKERNELBIN'
    ret

;*****
; PROSEDURE LBACHS
; convert 'ax' LBA addressing scheme to CHS addressing scheme
; absolute sector = (logical sector / sectors per track) + 1
; absolute head   = (logical sector / sectors per track) MOD number of heads
; absolute track  = logical sector / (sectors per track * number of heads)
;*****
LBACHS:
    xor     dx, dx            ; prepare dx:ax for operation
    div     WORD [SectorsPerTrack] ; calculate
    inc     dl                ; adjust for sector 0
    mov     BYTE [absoluteSector], dl
    xor     dx, dx            ; prepare dx:ax for operation
    div     WORD [NumHeads]    ; calculate
    mov     BYTE [absoluteHead], dl
    mov     BYTE [absoluteTrack], al
    ret
;*****

```

```
bootasm - Notepad
File Edit Format View Help

mov     BYTE [absoluteHead], dl
mov     BYTE [absoluteTrack], al
ret

;=====
; Definisi VARIABEL umum
;=====
absoluteSector db 0x00
absoluteHead   db 0x00
absoluteTrack  db 0x00

datasetector  dw 0x0000
cluster       dw 0x0000

;=====
; Definisi NAMA file OS/Kernel
; 11 Karakter: Terdiri dari
; 8 karakter untuk nama file,
; 3 Karakter untuk ekstensi
; Tanpa TITIK '.' pemisah
; perintah 'dir' akan menampilkan 'KERNEL.BIN'
;=====
ImageName db "KERNEL BIN"

;=====
; Teks yang akan ditampilkan saat mulai proses
; BOOT : Loading kernel ver 0.01
; Di awali dan diakhiri dengan tanda :
; -- 0x0D akhir baris
; -- 0x0A baris baru
; -- 0x00 Karakter 'NULL' pembatas dengan data di bawahnya
;=====
msgLoading db 0x0D, 0x0A, "Loading kernel ver 0.01 ", 0x0D, 0x0A, 0x00
msgCRLF   db 0x0D, 0x0A, 0x00
msgProgress db ".", 0x00
```

```
bootasm - Notepad
File Edit Format View Help

;=====
; Definisi NAMA file OS/Kernel
; 11 Karakter: Terdiri dari
; 8 karakter untuk nama file,
; 3 Karakter untuk ekstensi
; Tanpa TITIK '.' pemisah
; perintah 'dir' akan menampilkan 'KERNEL.BIN'
;=====
ImageName db "KERNEL BIN"

;=====
; Teks yang akan ditampilkan saat mulai proses
; BOOT : Loading kernel ver 0.01
; Di awali dan diakhiri dengan tanda :
; -- 0x0D akhir baris
; -- 0x0A baris baru
; -- 0x00 Karakter 'NULL' pembatas dengan data di bawahnya
;=====
msgLoading db 0x0D, 0x0A, "Loading kernel ver 0.01 ", 0x0D, 0x0A, 0x00
msgCRLF   db 0x0D, 0x0A, 0x00
msgProgress db ".", 0x00

;=====
; Teks yang di tampilkan saat terjadi kesalahan BOOT
;=====
msgFailure db 0x0D, 0x0A, "ERROR : Press Any Key to Reboot", 0x00

;=====
;(4) BLOK BOOT SIGNATURE
;=====
TIMES 510-($-$$) DB 0
DW 0xAA55
;=====
```