

# **Tugas Besar Milestone 1 - Beneath The Skin Operating System Development**

IF2230 - Sistem Operasi

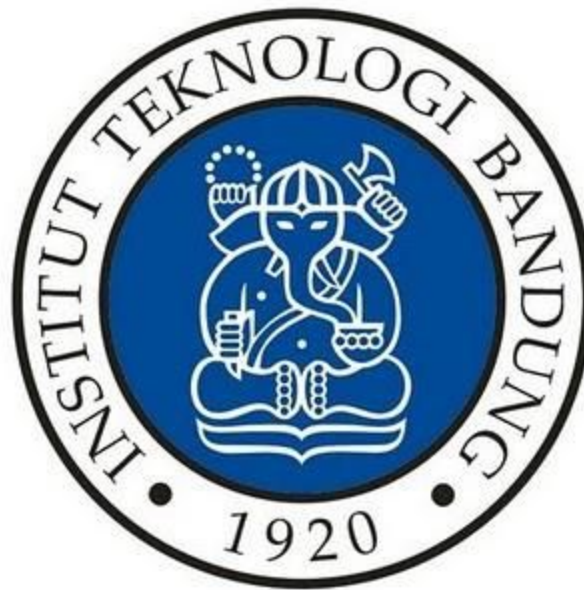
Diajukan sebagai tugas dari mata kuliah Sistem Operasi di jurusan Teknik Informatika Institut Teknologi Bandung

Oleh :

**Farhan Amin (13515043)**

**Dery Rahman A (13515097)**

**Aulia Ichsan Rifkyano (13515100)**



**PROGRAM STUDI INFORMATIKA  
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG  
2017**

## A. Pertanyaan

- a. **Apa perbedaan cara booting disk MBR dengan GPT? Apa cara boot yang dipakai dalam tugas ini?**

MBR (*Master Boot Record*) dan GPT (*Globally Unique Identifier Partition Table*) merupakan jenis struktur yang digunakan dalam *disk* untuk mengatur bagaimana partisi sebuah disk dilakukan. MBR menempatkan sektor bootnya hanya pada bagian awal *disk*, sedangkan GPT menempatkan sektor bootnya, selain pada *protective MBR*, lebih dari 1 kali dalam suatu disk. Sehingga ketika terjadi kerusakan pada sektor awal, GPT masih mempunyai cadangan untuk melakukan *recovery*. MBR sendiri mempunyai beberapa kelemahan, antara lain hanya mendukung *disk* dengan kapasitas maksimum 2TB saja. Selain itu, untuk membuat partisi lebih dari 4, diharuskan untuk membuat 3 partisi primer, dan 1 partisi *extended* yang berisi partisi-partisi *logic* lainnya. Kekurangan yang paling fatal adalah, lokasi partisi sistem hanya disimpan pada sektor awal saja, sehingga apabila hilang atau mengalami kerusakan maka komputer tidak bisa melakukan boot ke OS tersebut. Berbeda dengan GPT yang menempatkan *backup* sistemnya kedalam beberapa lokasi.

Cara boot yang dipakai dalam tugas ini adalah menggunakan MBR.

- b. **Apakah mungkin menginstall OS ini ke floppy disk atau disk lain? Jelaskan.**

Bisa. File *floppya.img* pada dasarnya merupakan file bootable yang sama seperti *bootable devices* OS lainnya. Kami mencoba menggunakan VMWare untuk melakukan testnya, dan berhasil dijalankan, seperti yang terlihat di Gambar 4. Pada awal PC dinyalakan, BIOS akan mengetes dan akan melakukan *booting* yang sebenarnya. Proses ini disebut *power on self test* (POST). Sistem akan memilih *disk* yang mengandung *bootable file*, pertama mencari didalam sektor awal *floppy drive*, jika tidak ditemukan (tidak ada *floppy disk* yang dimasukkan), sistem akan mencari didalam sektor pertama *hard disk*. *Boot sector* ini bisa disebut *master boot record* dalam *hard disk*, karena dalam *hard disk* dapat mengandung beberapa partisi, dan memiliki *boot sector* sendiri.

- c. **Apa perbedaan booting secara real mode dan protected mode?**

*Booting* secara *real mode* berarti *booting* pada prosesor x86 yang memberikan program akses langsung ke memori, *I/O Address*, dan *interrupts*. Memori yang tersedia pada *real mode* sebesar 1MB dari RAM dengan 20bit ruang pemakaian untuk setiap alamat memorinya. Pada *real mode* tidak ada proteksi karena langsung menggunakan alamat fisik memori (tidak terdapat virtual memori), tidak terdapat perlindungan kepada memori, tidak dimungkinkan *multitasking*, dan tidak ada pengaturan hak akses kepada suatu program terhadap *resource* yang tersedia (semua milik bersama). Program 16bit lah yang dapat dijalankan pada mode ini. Sedangkan untuk *Booting* secara *protected mode* adalah *booting* pada prosesor x86 yang menyediakan fitur-fitur yang tidak disediakan pada *real mode* yakni, pengaturan hak akses program terhadap *resource*, *virtual memory*, *paging*, dan *multitasking* yang mampu meningkatkan kontrol sistem operasi terhadap program. Mode ini dapat menjalankan program 32bit maupun 16bit.

Letak perbedaan *real mode* dan *protected mode* terdapat pada perlindungannya. Pada *real mode booting* dapat mengakses seluruh *hardware* secara bebas dan mengakses memori secara langsung, dengan cara seperti ini, ada kemungkinan bahwa *hardware* dan *memori* menjadi *fragile* dan dapat berbahaya. Sedangkan pada *protected mode* tiap program diberikan hak akses terhadap *hardware* secara terbatas dan pengaksesan memori dilakukan melalui virtual memori sehingga bisa menghilangkan faktor *hardware* dan *memory* yang secara langsung terpapar yang dapat membahayakan komputer. Pada *real mode* memori yang dapat digunakan hanya 1MB dari RAM,

dan pada mode ini juga *multitasking* tidak dapat dilakukan. Sedangkan pada *protected mode*, memori yang dapat digunakan lebih dari 1MB dan *multitasking* dapat dilakukan.

d. **Mode apa yang dipakai dalam tugas ini?**

Mode booting yang dipakai dalam tugas ini adalah real mode. Hal ini dapat dilihat pada program yang dibuat yaitu, kernel.c yang memanfaatkan fungsi putInMemory yang mengakses alamat fisik memori pada RAM (tidak menggunakan memori virtual).

e. **Mengapa BIOS sekarang masih ada yang dieksekusi secara 16-bit?**

Karena sejarahnya. BIOS adalah suatu hal yang bekerja ketika sebuah komputer dinyalakan. Dengan begitu, processor pun harus berada di suatu mode agar bisa membuat BIOS bekerja. Saat pertama kali dibuat, sebuah PC memiliki processor type 8088, type serupa dari 8086 processor yang ada dengan 8-bit external data bus. Walaupun 8086/8088 adalah processor 16-bit, mereka didesain untuk menjadi *assembly-language compatible* dengan keluarga processor 8088 8-bit sebelumnya untuk membuat port existing software ke dalam chip baru mudah. Hal ini lalu menjadi tradisi: setiap chip baru akan berusaha keras untuk tetap compatible dengan processor sebelum-sebelumnya.

Saat processor 386 keluar, yaitu processor 32-bit pertama di keluarga x86, para desainer processor pun memiliki masalah: processornya harus bisa menjalankan program 16-bit (yang berarti *assembly-language compatibility* terhadap 8080) yang sudah ada. Solusinya adalah membuat sebuah *processor* yang bekerja mulai dengan mode 16-bit compatibility dan membuat sebuah software yang dapat mengubah modenya menjadi 32-bit. Dengan menggunakan mode 16-bit, maka *processor* dengan bit yang lebih tinggi dapat menjalankan program-program 16-bit yang dibutuhkan, hal ini disebut *backward compatibility*.

Hingga saat ini, banyak processor designer yang mengira bahwa hal ini bekerja dan bukan sebuah masalah besar, oleh karena itu saat ini masih banyak yang BIOS di dalam komputernya dieksekusi dalam mode 16-bit.

f. **Mengapa magic number yang digunakan oleh bootloader adalah 55 AA? Apa akibatnya jika magic number tersebut diubah nilainya?**

Karena 55 AA atau AA 55 merupakan sebuah pattern unik yaitu 1010101001010101 (AA 55) yang bisa membedakan komputer yang mana menggunakan little endian dan komputer mana yang menggunakan big endian. Jika nilainya diubah kemungkinan besar akan menjadi error atau menampilkan "bootable disk not found"

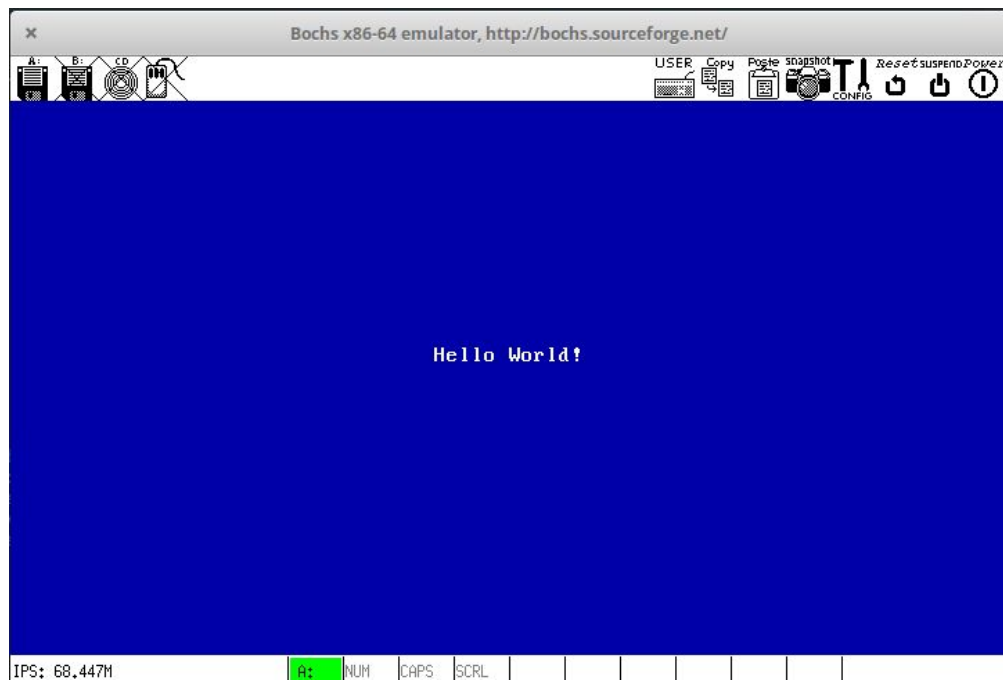
## B. Kesulitan saat mengerjakan

Pada awalnya, kami mengalami kesulitan ketika mencoba untuk mengcompile kernel.c, hal ini dikarenakan kami melakukan implementasi tanpa menggunakan main (langsung memanggil fungsi putInMemory). Setelah kami analisis, kami mengerti bahwa fungsi putInMemory tersebut perlu diletakkan didalam main saat pemanggilannya.

Selanjutnya, kami tidak mengalami kesulitan dalam membuat dan mengimplementasikan beberapa fungsi lain kedalam kernel.c, baik fungsi printHelloWorld, printString, maupun printLogo. *Boot* berjalan sesuai dengan harapan kami.

Ketika kami mencoba untuk mengimplementasikan *integer* dalam kernel.c, tipe tersebut hanya dapat mendukung *value* hingga 65,535 ( $2^{16} - 1$ ). Hal ini karena sistem 16bit hanya dapat menampung  $2^{16}$  *value* saja pada registernya, berbeda dengan sistem 32bit yang dapat menampung hingga  $2^{32}$  *value*.

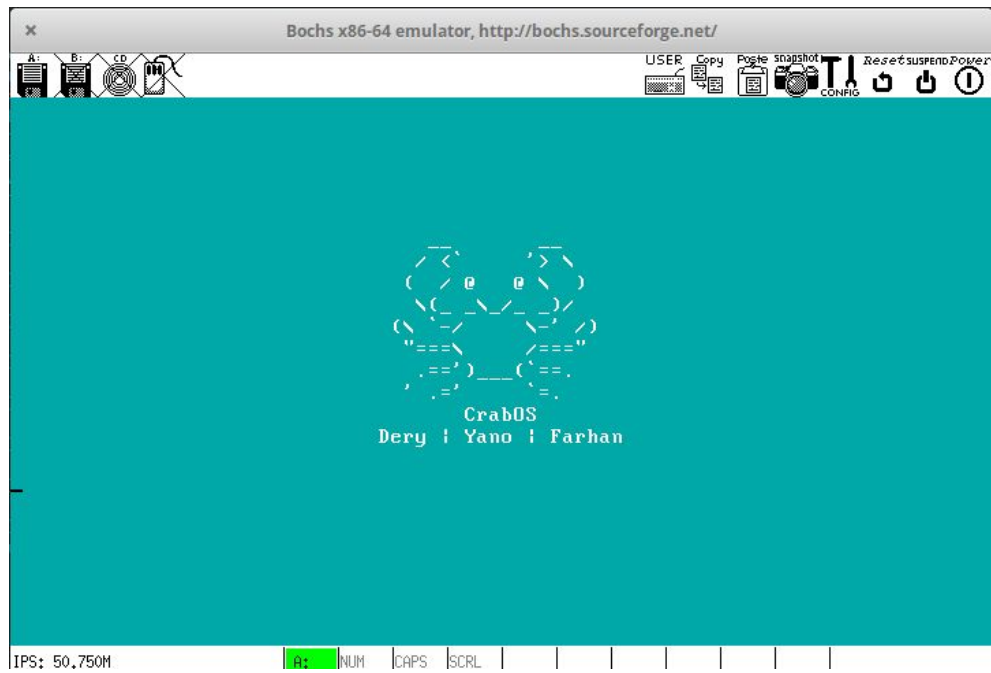
### C. *Screenshot*



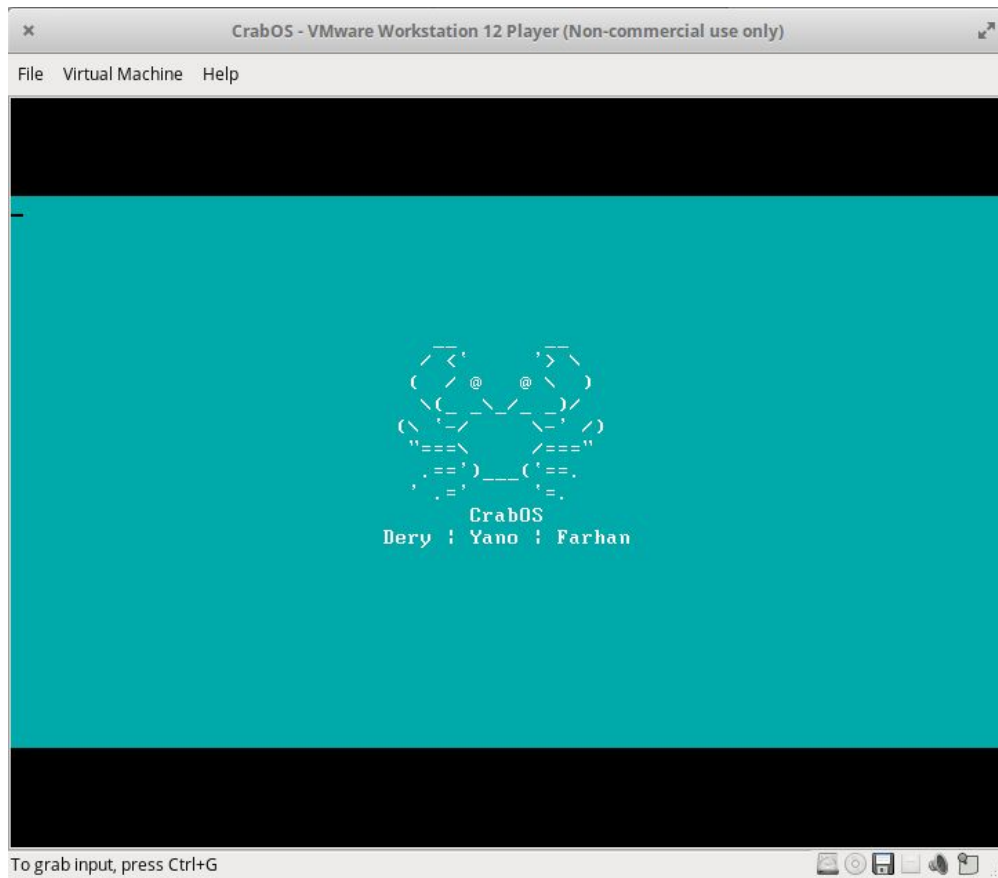
Gambar 1. Menampilkan Hello World



Gambar 2. Menampilkan text dengan fungsi printString



Gambar 3. Menampilkan logo Kepiting



Gambar 4. Menjalankan dengan VMWare

## D. Pembagian tugas

Nama / NIM	Yang dikerjakan	Perkiraan persentase
Farhan Amin / 13515043	<ul style="list-style-type: none"><li>• Mengerjakan soal c dan d</li><li>• Mengerjakan Hello World</li></ul>	30%
Dery Rahman A / 13515097	<ul style="list-style-type: none"><li>• Membuat fungsi printString</li><li>• Membuat fungsi printLogo</li><li>• Membuat fungsi clearScreen</li><li>• Mengerjakan soal a dan b</li></ul>	40%
Aulia Ichsan Rifkyano / 13515100	<ul style="list-style-type: none"><li>• Mengerjakan soal e dan f</li><li>• Mengerjakan printString</li></ul>	30%

## E. Feedback

Deskripsi dan langkah-langkah pengerjaan tugas mudah dipahami, karena tersusun secara sistematis. Pertanyaan-pertanyaan yang diberikan sudah cukup baik, sehingga membuat mahasiswa dapat mengeksplorasi lebih dalam lagi tentang permasalahan yang diberikan.

Source:

<http://www.tldp.org/LDP/sag/html/boot-process.html>

<https://www.howtogeek.com/193669/whats-the-difference-between-gpt-and-mbr-when-partitioning-a-drive/>

<https://www.quora.com/Why-does-BIOS-operate-in-16-bits-instead-of-32-64-bits>

<http://stackoverflow.com/questions/1125025/what-is-the-role-of-magic-number-in-boot-loading-in-linux>

<http://minnie.tuhs.org/cgi-bin/utree.pl>

<http://mbrwizard.com/thembr.php>