

# The BIOS and Boot Process

Haidar Tafazul Fikri (24051130004)

Yohanes Joni Kurniawan (24051130018)

Mohamad Rikza Sholihin (24051130027)

Gayuh Aza (24051130032)

## 2.1 BIOS Actions

Secara singkat BIOS (*Basic Input Output System*) merupakan firmware penting yang bertanggung jawab terhadap langkah-langkah awal dalam proses booting komputer. BIOS tertanam dalam ROM (*Read-Only Memory*) sehingga bersifat non-volatile, artinya isinya tidak akan hilang meskipun komputer dimatikan. BIOS memiliki fungsi utama yaitu mempersiapkan lingkungan sistem agar sistem operasi bisa dijalankan, dengan cara menginisialisasi komponen perangkat keras, menjalankan pengujian awal, serta memulai proses booting dari media penyimpanan yang dapat di-boot (*bootable device*). BIOS juga memberikan antarmuka konfigurasi kepada pengguna melalui CMOS Setup Utility.

Secara umum, langkah-langkah awal yang dilakukan BIOS bisa dibagi menjadi tiga bagian utama: *Power-On Self Test* (POST), menampilkan logo pabrikan BIOS sebagai tanda keberhasilan inisialisasi, dan penyediaan akses ke pengaturan konfigurasi sistem melalui CMOS.

### 2.1.1 Power-On Self Test (POST)

POST merupakan proses pengujian perangkat keras yang dijalankan secara otomatis segera setelah komputer dinyalakan. Tujuan utamanya adalah memastikan bahwa komponen inti dari sistem, seperti RAM, *processor*, keyboard, dan media penyimpanan terhubung dengan baik dan dalam kondisi yang memungkinkan untuk melanjutkan ke proses berikutnya.

Jika semuanya berjalan dengan baik, proses ini biasanya hanya berlangsung beberapa detik dan pengguna tidak menyadari secara langsung. Namun, apabila ditemukan masalah, sistem akan memberikan peringatan berupa *beep code*, yaitu serangkaian bunyi pendek atau panjang dari speaker internal. Kode ini berbeda tergantung pada jenis BIOS yang digunakan

(misalnya AMI BIOS, Award BIOS, dll), dan dapat digunakan untuk mendiagnosis masalah spesifik seperti kerusakan RAM, kegagalan VGA, atau masalah pada keyboard.

Yang menarik, karena sistem operasi belum berjalan dan sistem audio belum aktif, sinyal bunyi tersebut berasal dari speaker kecil internal (biasanya piezo buzzer) yang tertanam di motherboard, bukan dari speaker utama yang biasa digunakan untuk multimedia.

POST juga memverifikasi apakah perangkat penyimpanan seperti harddisk, SSD, atau optical drive dapat digunakan sebagai sumber untuk mem-boot sistem operasi. Jika tidak ditemukan perangkat bootable, BIOS akan menghentikan proses dan menampilkan pesan error di layar seperti:

**"No bootable device found – insert system disk and press any key."**

### **2.1.2 BIOS Manufacturer's Logo**

Setelah POST selesai dan perangkat keras terdeteksi dalam kondisi siap, BIOS menampilkan logo atau splash screen dari pabrikan BIOS. Ini merupakan langkah visual yang menunjukkan bahwa sistem siap melanjutkan ke proses booting. Merek-merek BIOS yang umum dijumpai di pasaran antara lain:

- Phoenix BIOS
- AMI (American Megatrends Inc.) BIOS
- Award BIOS
- Intel BIOS
- IBM BIOS

Logo ini biasanya disertai dengan informasi teknis seperti versi BIOS, tombol untuk masuk ke menu konfigurasi (contohnya: "Press DEL to enter setup" atau "F2 for BIOS Setup"), dan daftar perangkat penyimpanan yang terdeteksi.

Walaupun tampilan ini hanya berlangsung beberapa detik, pada dasarnya ini merupakan antarmuka penting yang menandai transisi dari tahap inisialisasi ke tahap pemuatan sistem operasi.

### 2.1.3 CMOS dan Setup Modifications

CMOS (Complementary Metal-Oxide Semiconductor) dalam konteks BIOS mengacu pada memori kecil yang digunakan untuk menyimpan konfigurasi sistem. CMOS bersifat *read/write*, artinya penggunaannya dapat mengubah isinya melalui menu BIOS Setup Utility. Berbeda dari ROM BIOS yang hanya bisa dibaca, CMOS dapat diubah dan menyimpan data konfigurasi seperti:

- Tanggal dan waktu sistem
- Jenis dan kapasitas drive (HDD, SSD, CD/DVD)
- Urutan perangkat boot (boot sequence)
- Pengaturan daya (power management)
- Aktivasi/deaktivasi port (USB, serial, parallel)
- Password BIOS (untuk keamanan)
- Pengaturan kipas, suhu, dan fitur-fitur spesifik motherboard modern

CMOS ditenagai oleh baterai kecil (biasanya CR2032) yang menempel di motherboard. Baterai ini menjaga agar konfigurasi tidak hilang saat komputer dimatikan. Jika baterai habis, BIOS akan kembali ke pengaturan default dan waktu sistem biasanya akan kembali ke nilai awal (sering tahun 2000 atau 1980-an).

Selain itu, CMOS memiliki *checksum* kode verifikasi berdasarkan isi CMOS. Jika sistem mendeteksi checksum tidak sesuai (karena data korup atau baterai habis), BIOS akan

memberikan peringatan seperti:

**"CMOS checksum error – Defaults loaded."**

Pengguna kemudian dapat masuk ke setup, mengatur ulang konfigurasi, dan menyimpannya kembali agar digunakan dalam sesi berikutnya.

## 2.2 The Operating System Boot Process

Setelah BIOS menyelesaikan tugasnya, langkah berikutnya adalah memulai proses booting sistem operasi. Sistem operasi, yang merupakan perangkat lunak besar dan kompleks, tidak dapat disimpan seluruhnya di dalam ROM. Oleh karena itu, BIOS hanya bertanggung jawab untuk memuat bagian pertama dari sistem operasi, yaitu *bootstrap loader*.

Bootstrap loader adalah program kecil, biasanya berukuran 512 byte, yang berada di sektor pertama (track 0, sektor 0) dari media penyimpanan yang bisa di-boot seperti hard disk, SSD, CD/DVD, atau USB flash drive. BIOS memuat program ini ke lokasi tertentu di RAM dan kemudian menyerahkan kendali sistem kepada bootstrap.

Bootstrap memiliki peran penting: memuat bagian utama dari sistem operasi, yaitu **kernel**, ke dalam memori. Kernel adalah inti sistem operasi yang menangani manajemen proses, manajemen memori, sistem file, dan komunikasi antar perangkat.

BIOS hanya mengetahui bagaimana membaca sektor awal dari media penyimpanan. Maka, tugas sebenarnya untuk memuat sistem operasi dalam bentuk utuh dilakukan oleh bootstrap. Proses ini biasanya berlanjut dengan:

1. Pemuatan kernel OS ke RAM
2. Inisialisasi subsistem OS (file system, input/output, network, dsb.)
3. Pemanggilan layanan sistem pertama, seperti login screen atau graphical desktop

Karena sistem bisa memiliki lebih dari satu media bootable (misalnya CD dan hard disk), BIOS menyediakan urutan prioritas boot yang bisa dikonfigurasi dalam setup. Jika perangkat pertama tidak memiliki sistem bootable, BIOS akan memeriksa perangkat berikutnya dalam daftar.

## 2.3 Protection Mechanisms

Dalam lingkungan sistem modern, perlindungan sistem menjadi sangat penting, khususnya saat lebih dari satu proses atau pengguna berjalan secara bersamaan. Tanpa perlindungan, satu proses bisa mengakses data proses lain, merusak OS, atau bahkan mengendalikan seluruh sistem secara ilegal. Maka, sistem operasi dan perangkat keras dirancang dengan berbagai *protection mechanisms*.

### Instruksi Terbatas (Privileged Instructions)

Tidak semua instruksi mesin boleh digunakan oleh sembarang program. Instruksi seperti menghentikan CPU (HLT), mengubah tabel interrupt, atau mengakses I/O langsung dikategorikan sebagai **privileged** dan hanya boleh dijalankan dalam mode kernel. Jika program biasa mencoba menjalankannya, maka sistem akan memicu *trap* (sejenis *interrupt*), dan biasanya proses tersebut akan dihentikan oleh OS.

### User Mode vs Kernel Mode

Mode operasi komputer dibagi dua:

- **User Mode:** untuk menjalankan aplikasi biasa, dengan akses terbatas
- **Kernel Mode:** untuk menjalankan OS dan kernel-level routines, dengan akses penuh

Transisi dari user mode ke kernel mode hanya terjadi saat program memanggil system call, misalnya untuk membaca file atau mencetak ke layar. Setelah fungsi kernel selesai, kontrol kembali ke user mode.

### Proteksi Memori dan Akses

Setiap proses hanya boleh mengakses ruang memorinya sendiri. Sistem operasi, dibantu oleh hardware seperti MMU (*Memory Management Unit*), memastikan proses tidak bisa membaca atau menulis ke memori proses lain. Ini mencegah terjadinya pelanggaran hak akses yang dapat menyebabkan sistem crash atau kebocoran data.

## 2.4 Think of Making Your Own Operating System

Untuk membantu memahami secara praktis bagaimana proses boot dan kendali sistem bekerja, penulis mendorong pembaca untuk membuat *sistem operasi mini* mereka sendiri. Walaupun tidak fungsional sebagai OS sungguhan, ini memberi wawasan tentang bagaimana bootstrap dan BIOS bekerja.

Langkah-langkah yang disarankan:

1. **Buat program sederhana** (misalnya mencetak teks “Hello”) dalam bahasa C atau Assembly.
2. **Hapus header executable-nya**, sehingga hanya tersisa instruksi mentah.
3. **Simpan ke sektor pertama** dari sebuah disket (track 0, sektor 0).
4. **Set BIOS untuk boot dari floppy**.
5. **Jalankan komputer**, dan sistem akan menjalankan program buatanmu sebagai OS.

Walaupun tidak bisa menjalankan aplikasi lain, OS buatan sendiri ini sudah memiliki kendali penuh atas CPU dan memori. Ini adalah pengalaman langsung bagaimana OS mengambil alih kendali dari BIOS setelah proses booting.

## 2.5 Summary

Bab ini menjelaskan dengan sangat jelas bagaimana BIOS memulai siklus hidup sistem komputer. BIOS melakukan pengecekan awal, menampilkan antarmuka konfigurasi, dan memuat program kecil (bootstrap) dari media penyimpanan. Bootstrap kemudian memuat kernel sistem operasi, yang mengambil alih kontrol penuh atas sistem. Sistem operasi modern disusun secara bertingkat, mulai dari BIOS, kernel, hingga aplikasi dan antarmuka pengguna. Perlindungan terhadap memori, akses perangkat keras, dan kontrol mode eksekusi sangat penting untuk menjamin keamanan dan stabilitas sistem. Bab ini juga mendorong pembaca untuk secara eksploratif membuat OS mini untuk memperdalam pemahaman teknis secara langsung.