

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini, teknologi mikro komputer berkembang sangat pesat, demikian pula dengan pemanfaatannya yang semakin beraneka ragam. Kemajuan teknologi tersebut dimanfaatkan untuk memenuhi kebutuhan manusia, seperti halnya mikrokontroler yang sering dijumpai dalam sistem pengaturan kerja berbagai alat. Selain itu, manfaat kehadiran komputer juga sudah dirasakan oleh sebagian besar manusia di dalam berbagai pekerjaan rutin mereka tidak hanya di kantor, namun juga di rumah.

Sekalipun demikian, mikrokontroler tidak senantiasa dapat memenuhi kebutuhan manusia yang semakin bertambah kompleks. Contohnya dalam implementasi pada robot yang digunakan untuk melakukan pekerjaan yang kompleks. Keterbatasan memori dan daya proses jumlah data menjadikan mikrokontroler kurang memadai untuk pekerjaan yang kompleks. Beruntung masalah ini dapat terpecahkan dengan penggunaan prosesor dan memori sebagaimana terdapat pada komputer pribadi yang lazim digunakan di perkantoran atau rumah.

Dibandingkan mikrokontroler, kemampuan prosesor untuk memproses data berkali lipat lebih baik, karena didukung kapasitas memori hingga gigabytes. Tidak seperti pada mikrokontroler, yang kapasitas memorinya hanya mencapai beberapa kilobytes saja.

Prosesor 80x86 seperti pada Intel dan AMD sesungguhnya memiliki dua mode, yakni *real mode* dan *protected mode*. [7] Dalam *real mode* kapasitas maksimum memori yang dapat diakses hanya 1024KB saja, di dalam lingkungan 16 bit. Hal ini mengakibatkan terbatasnya kapabilitas prosesor untuk memenuhi kebutuhan program aplikasi yang cenderung semakin besar, tidak saja dalam ukuran kapasitas memori, tetapi juga penggunaan berbagai sumber daya sistem lainnya. Sementara pada *protected mode*, program memiliki keleluasaan akses untuk penggunaan memori yang lebih besar, yaitu sekurang-kurangnya 1024KB, di dalam lingkungan 32 bit, sehingga kemampuan prosesor dalam memenuhi kebutuhan program aplikasi dapat meningkat. Karena keunggulan inilah maka perlu dibuat suatu cara untuk merealisasikan perpindahan modus prosesor dari *real mode* ke *protected mode*.

1.2 Perumusan Masalah

Pada *real mode* kapasitas memori yang dapat diakses hanya mencapai 1024KB dan hanya bekerja pada lingkungan 16 *bit*. Dengan sifat dasar seperti ini kapabilitas prosesor menjadi terbatas untuk dapat memenuhi kebutuhan program aplikasi yang kecenderungannya semakin besar tidak saja dalam ukuran memori tetapi penggunaan berbagai sumber daya sistem lainnya. Pada *protected mode*, program memiliki keleluasaan akses bagi penggunaan ke memori yang lebih besar, sedikitnya 1024KB, di dalam lingkungan 32 *bit*. [1]

Dengan keuntungan yang dapat diperoleh ketika bekerja di dalam *protected mode*, maka adalah menarik untuk merealisasi sebuah pemikiran untuk

mewujudkan program perpindahan dari *real mode* ke *protected mode*, dan sebaliknya. Secara teoritis, transisi modus ini dapat dilakukan dengan mengubah *bit* PE yang terdapat pada *Control Register 0* (CR0) menjadi 1. Akan tetapi untuk dapat melakukan pekerjaan di *protected mode* dibutuhkan pendeklarasian *Global Descriptor Table* (GDT), *Interrupt Descriptor Table* (IDT), dan *Task State Segment* (TSS), pembuatan *driver* untuk *device*, pengaturan *multitasking* yang ada sedemikian rupa sehingga prosesor dapat mendukung kebutuhan program untuk menuntaskan pekerjaannya. Dalam penelitian ini, sebuah aplikasi utilitas dikembangkan untuk menjelaskan bagaimana proses perpindahan modus dapat dilakukan. [14]

1.3 Pembatasan Masalah

Pembahasan dalam penelitian dibatasi pada proses perpindahan dari *real mode* ke *protected mode* meliputi deklarasi GDT, IDT dan TSS sehingga sebuah program sederhana dapat dieksekusi pada lingkungan *protected mode*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah membuat program *assembly* untuk melakukan perpindahan mode dari *real mode* ke *protected mode* dan sebaliknya. Setelah melakukan perpindahan, program akan menjalankan sebuah prosedur yang dijalankan pada *protected mode* untuk menunjukkan bahwa sistem sudah berada pada *protected mode*.

1.5 Sistematika Penulisan

Penulisan Laporan Tugas Akhir meliputi pembagian bab sebagai berikut:

Bab I Pendahuluan

Berisi latar belakang masalah, perumusan masalah, pembatasan masalah, tujuan penelitian, dan sistematika penulisan.

Bab II Landasan Teori

Bab ini mencakup teori-teori mengenai arsitektur , metode-metode, prinsip-prinsip ataupun informasi lain yang dibutuhkan guna memecahkan masalah serta menyusun Laporan Tugas Akhir.

Bab III Perancangan Program

Bab ini menguraikan langkah-langkah yang digunakan untuk memecahkan masalah atau kerangka berpikir jalannya penelitian untuk menyelesaikan masalah. Uraian akan mencakup perancangan program untuk melakukan perpindahan *mode*.

Bab IV Implementasi Program

Bab ini berisi pembahasan mengenai program yang dibuat dan implementasinya

Bab V Kesimpulan dan Saran

Bab ini berisi kesimpulan dari Laporan Tugas Akhir berdasarkan pembahasan masalah dan analisis hasil. Dan juga saran untuk penelitian lebih lanjut.