

## **PROTEKSI MEMORI DAN CPU TERHADAP KESALAHAN PROGRAM PADA SISTEM OPERASI**

**Septilia Arfida**

Dosen pada Jurusan Teknik Informatika,  
Informatics & Business Institute Darmajaya  
Jl. Z.A Pagar Alam No 93, Bandar Lampung - Indonesia 35142  
Telp. (0721) 787214 Fax. (0721)700261  
Email: *septiliatime@gmail.com*

### **ABSTRACT**

*Operating system has began to divides the system resources to many programs simultaneously to improve system performance. But when the system runs without resource sharing, an error in the program will only cause problems in the program. Along with development of the operating system, then the overall control system is given to the operating system. In general, the memory need to be protected for the operating system accesses protected from user programs and also to protect user programs with one another. The program – a program the user can access directly to the Central Processing Unit to execute intructions. When a program error appears, the operating system must abnormally terminate the program, provide appropriate error messages, and frees memory used by the program.*

**Keywords:** *Memory Protection, Protection of Central Processing Unit, Operating System*

### **ABSTRAK**

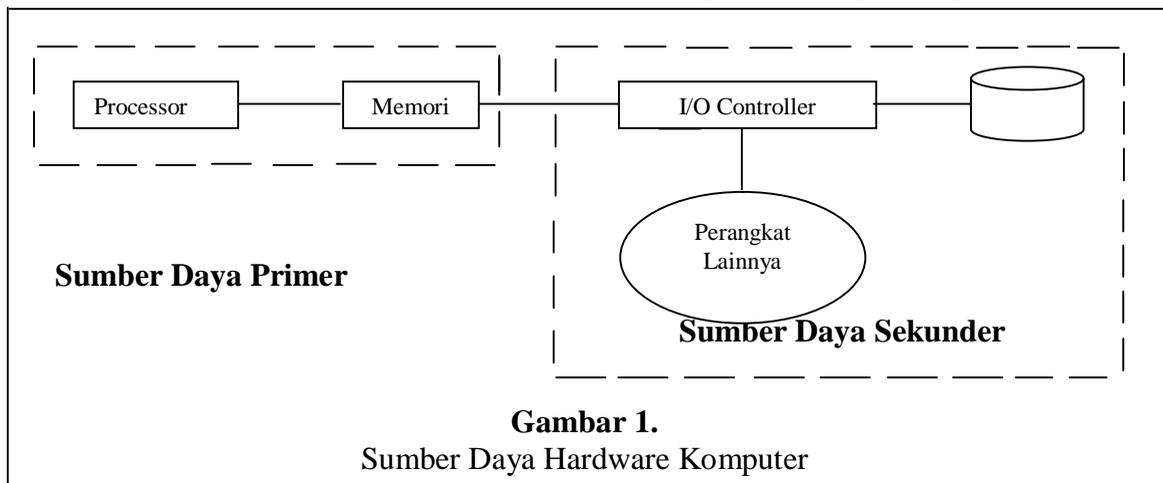
*Sistem operasi telah mulai membagi sumber daya sistem untuk banyak program secara bersamaan untuk meningkatkan kinerja sistem. Tapi ketika sistem berjalan tanpa berbagi sumber daya, kesalahan dalam program ini hanya akan menimbulkan masalah dalam program. Seiring dengan perkembangan sistem operasi, maka sistem kontrol secara keseluruhan diberikan kepada sistem operasi. Secara umum, memori perlu dilindungi untuk sistem operasi pengaksesan dilindungi dari program pengguna dan juga untuk melindungi program-program pengguna dengan satu sama lain. Program - program pengguna dapat mengakses langsung ke Central Processing Unit untuk mengeksekusi intructions. Ketika kesalahan program muncul, sistem operasi normal harus mengakhiri program, memberikan pesan kesalahan yang sesuai, dan membebaskan memori yang digunakan oleh program.*

**Kata kunci:** *Memory Protection, Perlindungan Central Processing Unit, Sistem Operasi*

## 1. PENDAHULUAN

Sistem operasi adalah sebuah program yang bertindak sebagai perantara antara pemakai komputer dan *hardware* komputer yang menyediakan suatu lingkungan di mana pemakai dapat mengeksekusi program. Sasaran utamanya

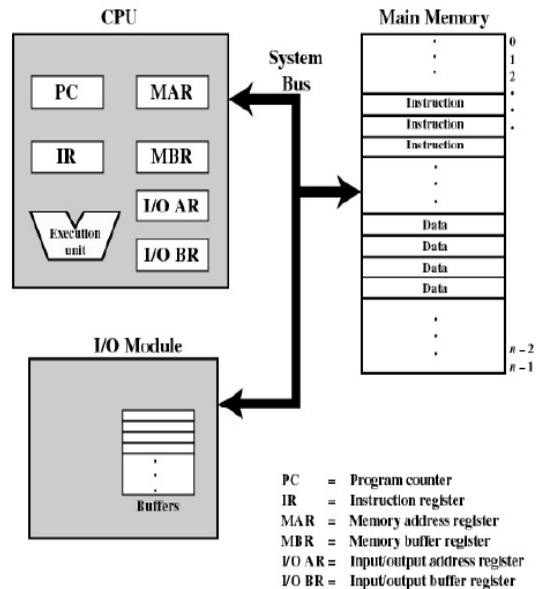
-adalah membuat sistem komputer digunakan secara tepat dan menggunakan hardware komputer seefisien mungkin. Sistem operasi dapat disebut sebagai *resource allocator* atau *resource manager* yang bertugas untuk mengalokasikan sumber daya komputer. Berikut gambaran tentang sumber daya komputer:



**Gambar 1.**  
Sumber Daya Hardware Komputer

Sistem komputer pada tingkat hardware terdiri dari empat komponen dasar yaitu Prosesor, Memori utama, Modul I/O, Sistem Bus. Prosesor berfungsi untuk mengontrol operasi-operasi komputer dan melakukan fungsi-fungsi pemrosesan data. Memori utama berfungsi menyimpan data dan program. Biasanya bersifat *volatile* dan sering disebut dengan memori primer atau memori nyata (*real memory*). *Volatile* tidak dapat mempertahankan data dan program yang disimpan bila sumber

daya energy (listrik) dipadamkan. Modul I/O berfungsi untuk memindahkan data antara komputer dan perangkat eksternal yang dibutuhkan. Perangkat eksternal biasanya terdiri dari perangkat memori sekunder, perangkat komunikasi, maupun terminal. Sistem Bus terdiri dari beberapa struktur dan mekanisme yang berguna untuk komunikasi antara prosesor, memori utama, dan modul I/O. Sistem komputer dengan empat komponen dasar dapat digambarkan sebagai berikut:



**Gambar 2.**

Empat Komponen dasar sistem komputer

Keterangan:

- PC (Program Counter), program terletak dalam memori kerja pada alamat memori tertentu. PC menerima alamat memori awal saat diolah oleh prosesor dan juga menyaring dan menentukan alamat memori mana saja yang isinya akan dibawa ke prosesor dan hanya alamat yang sesuai dengan letak program yang sedang diolah yang diperbolehkan PC masuk ke prosesor. PC mencacah dirinya sebesar 1 cacahan dan dengan cara yang sama akan melayani alamat berikutnya sampai alamat terakhir.
- IR (Instruction Register), membawa dan menerima informasi dari memori kerja dan diperiksa.
- MAR (Memory Address Register), Register ini untuk mencatat alamat memori yang akan diakses (baik yang akan dituliskan maupun dibaca).
- MBR (Memory Buffer Register), Register ini untuk menampung data yang akan dituliskan ke memori yang alamatnya ditunjuk MAR atau untuk menampung data dari memori (yang alamatnya ditunjuk oleh MAR) yang akan dibaca.
- I/O AR (Input Output Address Register), Register ini untuk mencatat alamat port I/O yang akan diakses (baik akan dituliskan maupun dibaca).
- I/O BR (Input Output Buffer Register), Register untuk menampung data yang akan dituliskan ke port yang alamatnya ditunjuk I/O AR atau

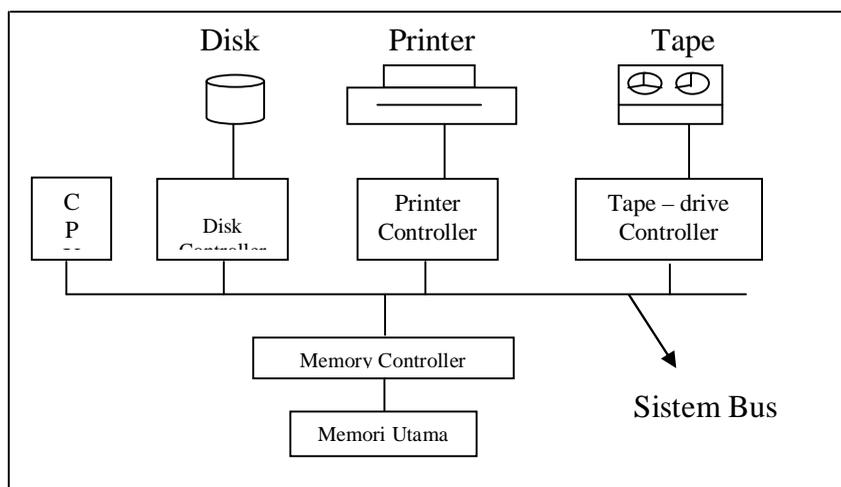
untuk menampung data dari port (yang alamatnya ditunjuk oleh I/O AR) yang akan dibaca.

## 2. PEMBAHASAN

Sistem komputer pada awalnya merupakan *single-user programmer operated system*, yang berarti bahwa pengoperasian komputer dilakukan oleh seorang programmer yang juga merupakan pemakai tunggal. Ketika programmer mengoperasikan komputer dari console, berarti programmer dapat mengontrol keseluruhan sistem dan memiliki kuasa penuh terhadap sistem.. Seiring dengan perkembangan sistem operasi, maka control keseluruhan sistem diberikan kepada sistem operasi.

Sistem komputer modern pada perkembangannya terdiri dari CPU dan sejumlah pengontrol perangkat (*device controller*) yang dihubungkan dengan sistem bus biasa yang menyediakan akses ke memori yang dapat dibagi-pakai

(*share*). Setiap device controller secara spesifik mengontrol perangkatnya contoh disk drive, video display dan perangkat audio. CPU dan device controller dapat bekerja secara bersamaan dan saling bersaing untuk mendapatkan akses ke memori. Oleh sebab itu dibutuhkan sebuah memori controller yang bertugas melakukan sinkronisasi akses ke memori. Sinkronisasi diperlukan untuk menghindari terjadinya ketidak-konsistenan data akibat adanya akses data secara konkuren. Proses-proses disebut konkuren jika proses-proses itu ada dan berjalan pada waktu yang sama, proses-proses konkuren ini bisa bersifat independen atau bisa juga saling berinteraksi. Proses-proses konkuren yang saling berinteraksi memerlukan sinkronisasi agar terkendali dan juga menghasilkan output yang benar. Sistem komputer modern dapat digambarkan sebagai berikut:



**Gambar 3.** Sistem Komputer Modern

Keterangan:

- Memori berfungsi untuk menyimpan data dan program. Setiap kali pemroses melakukan eksekusi, pemroses melakukan pembacaan instruksi dari memori utama.
- Memori Controller, yang bertugas melakukan sinkronisasi akses ke memori.
- Disk Controller  
Sebuah chip yang menerjemahkan perintah ke dalam sebuah form yang dapat mengatur sebuah hard disk drive.
- Printer Controller  
Alat yang digunakan untuk menghubungkan beberapa komputer dengan sebuah atau beberapa printer, sehingga sebuah printer dapat dipergunakan secara bersama-sama secara otomatis atau dapat dipindahkan antara satu printer dengan printer yang lain.  
Printer Controller ini dikatakan intelligent karena memiliki CPU dan memori tersendiri yang akan mengatur penggunaan printer bersama-sama. Memori di printer controller berfungsi sebagai buffer data yang akan dikirimkan ke printer untuk dicetak.
- Tape drive controller, yang mengatur pemakaian tape drive.  
Sistem operasi merupakan sebuah program sistem yang berguna untuk mengoperasikan komputer. Tanpa

sistem operasi maka komputer hanya merupakan perangkat elektronik yang tidak berguna. Sistem operasi dibangun di atas *interface hardware* dan menyediakan interface antara hardware dengan program-program aplikasi.

Program-program pemakai dapat mengakses secara langsung ke CPU untuk mengeksekusi instruksinya. Jika suatu program pemakai mengalami perulangan tidak terbatas (*looping* abadi), maka program tersebut tidak akan membebaskan kembali CPU dan mengembalikan ke sistem operasi. Hal ini akan mengganggu kinerja sistem secara keseluruhan.

Memori juga perlu diproteksi supaya sistem operasi diproteksi dari akses-akses program pemakai dan juga untuk memproteksi program pemakai yang satu dengan yang lainnya. Proteksi ini seharusnya disediakan oleh hardware. Memori berfungsi untuk menyimpan data dan program. Setiap kali pemroses melakukan eksekusi, pemroses melakukan pembacaan instruksi dari memori utama. Supaya intruksi dapat dilakukan secara cepat maka harus diusahakan instruksi tersedia di memori pada hirarki berkecepatan akses lebih tinggi. Kecepatan eksekusi ini akan meningkatkan kinerja sistem. Memori utama biasanya bersifat *volatile* dan

sering disebut dengan memori primer atau memori nyata (real memory). Untuk memperbaiki unjuk kerja sistem, sistem operasi mulai membagi sumber daya sistem kepada banyak program secara bersamaan. Sistem pembagian ini menciptakan perbaikan unjuk kerja dan sekaligus menimbulkan persoalan karena ketika sistem berjalan tanpa pembagian sumber daya, suatu kesalahan di dalam program hanya akan menyebabkan masalah pada program tersebut. Tetapi dengan adanya pembagian sumber daya, banyak proses dapat dirugikan karena terjadinya suatu kesalahan di dalam suatu program.

Komputer harus mengeksekusi satu program pada satu saat saja atau semua output harus diduga, bila tanpa adanya proteksi terhadap kesalahan. Suatu sistem operasi yang dirancang dengan baik harus dapat menjamin bahwa program yang berjalan dengan tidak benar, tidak dapat menyebabkan program yang lain dieksekusi secara tidak benar. Banyak kesalahan pemrograman dideteksi oleh hardware. Kesalahan ini secara normal ditangani oleh sistem operasi. Ketika kesalahan program muncul, sistem operasi harus menghentikan secara tidak normal program tersebut, memberikan pesan

kesalahan yang tepat, dan membebaskan memori yang dipakai oleh program tersebut.

Secara umum, memori perlu diproteksi supaya sistem operasi diproteksi dari akses-akses program pemakai dan juga untuk memproteksi program pemakai yang satu dengan yang lainnya. Proteksi ini seharusnya disediakan oleh hardware. Memori yang dapat digunakan oleh program pemakai dimulai dari suatu alamat yang disebut dengan *base register*. Banyaknya memori yang akan dialokasikan untuk program pemakai disebut dengan *bound register*.

Program-program pemakai dapat mengakses secara langsung ke CPU untuk mengeksekusi instruksinya. Jika suatu program pemakai mengalami perulangan tidak terbatas (*looping* abadi), maka program tersebut tidak akan membebaskan kembali CPU dan mengembalikan ke sistem operasi. Hal ini akan mengganggu kinerja sistem secara keseluruhan sehingga sistem operasi menggunakan *timer* (clock). Clock disebut juga timer, sangatlah penting untuk sistem yang menggunakan time sharing. Clock akan memelihara waktu setiap harinya dan juga akan melindungi suatu proses dari

proses yang lainnya dalam hal monopoli penggunaan CPU.

Memori beraneka tipe dari yang tercepat aksesnya sampai yang terlambat. Tipe tercepat adalah Cache Memory dan Main Memory sedangkan untuk tipe terlambat adalah Sekunder Memory. Memori berfungsi untuk menyimpan data dan program. Setiap kali pemroses melakukan eksekusi, pemroses melakukan pembacaan instruksi dari memori utama. Agar intruksi dapat dilakukan secara cepat maka harus diusahakan instruksi tersedia di memori pada hirarki berkecepatan akses lebih tinggi. Kecepatan eksekusi ini akan meningkatkan kinerja sistem. Memori utama biasanya bersifat *volatile* dan sering disebut dengan memori primer atau memori nyata (*real memory*). Memori yang dapat digunakan oleh program pemakai dimulai dari suatu alamat yang disebut dengan *base register*. Banyaknya memori yang akan dialokasikan untuk program pemakai disebut dengan *bound register*. Ketika prosesor berada dalam monitor mode, semua alamat yang digunakan merupakan alamat memori fisik (alamat sebenarnya), yang berarti alamat yang dihasilkan oleh instruksi dikirimkan secara langsung ke memori

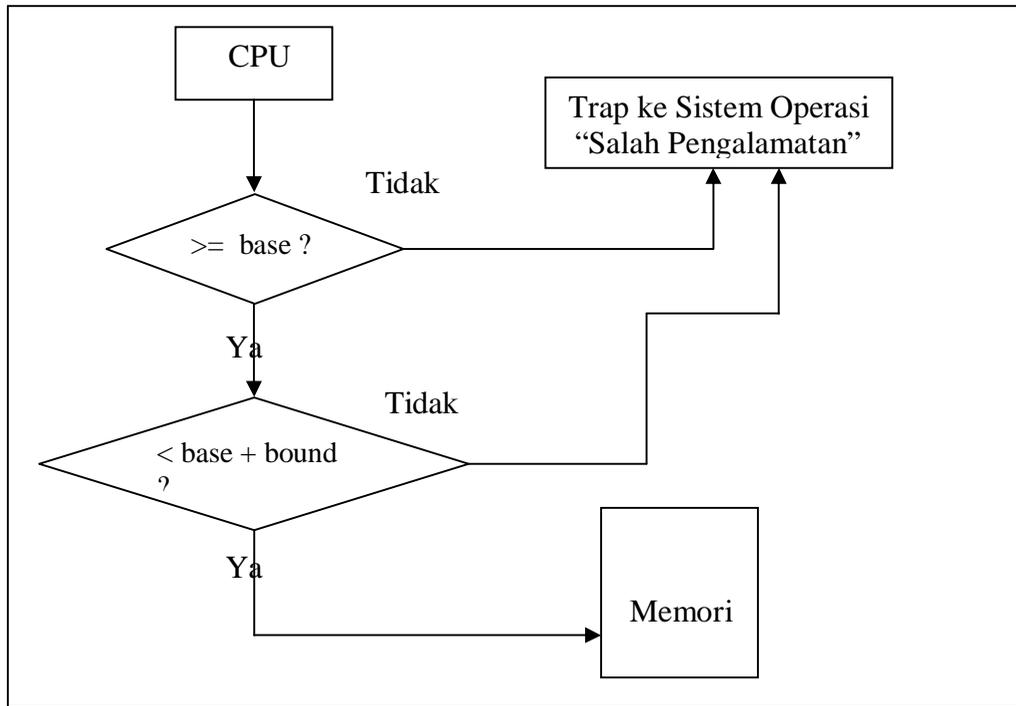
dengan menempatkan alamat tersebut ke *bus system* sebagai alamat memori. Alamat memori fisik seringkali disebut dengan *alamat absolute*.

Ketika prosesor berada dalam user mode, maka semua alamat yang digunakan merupakan alamat memori logik. Saat suatu instruksi menghasilkan sebuah alamat logik, prosesor akan memeriksa terlebih dahulu untuk meyakinkan bahwa alamat logic tersebut kurang dari hasil penjumlahan bound register dengan base register (bound register juga disebut sebagai *limit register*). Jika ternyata lebih besar, maka interrupt kesalahan program akan dihasilkan. Dari keterangan tersebut dapat ditarik kesimpulan bahwa dalam user mode, suatu program memiliki akses yang terbatas ke alamat fisik memori. Program tersebut tidak dapat mengakses sembarangan alamat fisik yang kurang dari nilai base register yang telah ditentukan, dan tidak dapat mengakses sembarangan alamat fisik yang lebih besar dari atau sama dengan hasil penjumlahan base register dengan bound register.

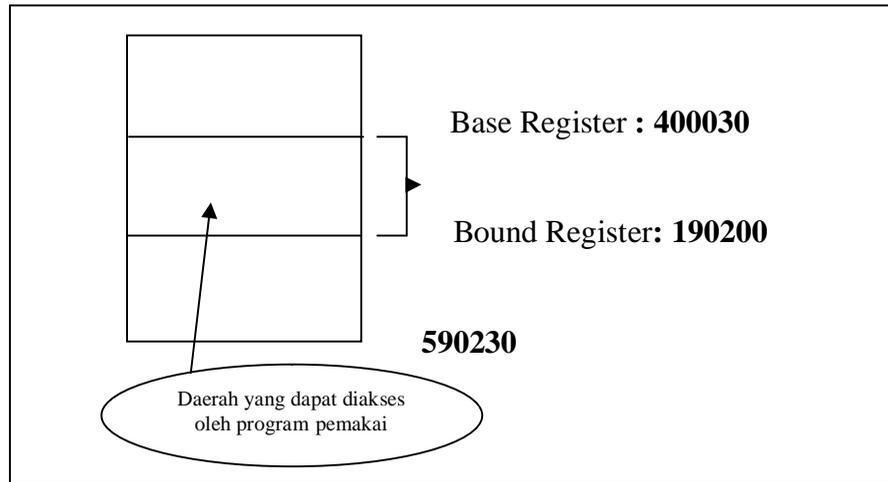
Sebagai contoh, jika base register berisi alamat 400030 dan bound register berisi 190200, maka program hanya dapat mengakses alamat memori secara

legal dari alamat 400030 sampai dengan 590230 (yang berasal dari penjumlahan 400030 dengan 190200).

Berikut tampilan flowchart proteksi memori dan Alamat memori yang dapat diakses oleh program pemakai:



**Gambar 4.** Flowchart Proteksi Memori



**Gambar 5.** Alamat Memori yang Dapat Diakses Oleh Program Pemakai

Program-program pemakai dapat mengakses secara langsung ke CPU untuk mengeksekusi instruksinya. Jika suatu program pemakai mengalami perulangan tidak terbatas (*looping* abadi), maka program tersebut tidak akan

membebaskan kembali CPU dan mengembalikan ke sistem operasi. Hal ini akan mengganggu kinerja sistem secara keseluruhan. Untuk mengatasi hal tersebut, sistem operasi menggunakan *timer*. Sebelum sistem operasi

memberikan CPU kepada program pemakai, sistem operasi akan mengeset timer untuk suatu periode waktu misalnya 20 minute per second. Jika program pemakai tidak mengembalikan CPU ke sistem operasi selama periode tersebut, maka timer akan menginterrupt program tersebut dan mengembalikan kontrol CPU ke sistem operasi.

Timer dapat ditetapkan untuk memberi interrupt pada periode waktu tertentu. Periode ini dapat bersifat tetap misal 1/60 detik atau berubah-ubah, misalnya dari 1 ms ke 1 s dengan kenaikan setiap 1 ms. Timer yang berubah-ubah (*variable timer*) ini biasanya diimplementasikan dengan menggunakan clock dengan kecepatan tertentu. Dan suatu pencacah (*counter*). Sistem Operasi menetapkan nilai pencacah tersebut. Setiap kali clock berdetak, maka nilai pencacah akan turun 1. Pada saat nilai pencacah sama dengan 0, maka akan muncul interrupt. Timer ini juga dapat digunakan untuk mencegah suatu program berjalan terlalu lama, dengan cara menetapkan lama waktu maksimum yang diijinkan oleh suatu program untuk *running*. Misal suatu program dengan batas waktu 7 menit, maka pencacah diinisialisasikan sama dengan 420. Setiap detik timer meng-interrupt dan nilai counter berkurang 1. Jika nilai pencacah positif, maka program

terus berjalan namun jika pencacah bernilai negative, maka system operasi akan menghentikan program karena telah melampaui batas waktu yang telah ditentukan.

### 3. KESIMPULAN

Sistem operasi pada dasarnya merupakan sebuah program sistem yang berguna untuk mengoperasikan komputer. Tanpa sistem operasi maka komputer hanya merupakan perangkat elektronik yang tidak berguna. Sistem operasi dibangun di atas *interface hardware* dan menyediakan interface antara hardware dengan program-program aplikasi. Untuk memperbaiki unjuk kerja sistem, sistem operasi mulai membagi sumber daya sistem kepada banyak program secara bersamaan. Sistem pembagian ini menciptakan perbaikan unjuk kerja dan sekaligus menimbulkan persoalan karena ketika sistem berjalan tanpa pembagian sumber daya, suatu kesalahan di dalam program hanya akan menyebabkan masalah pada program tersebut. Tetapi dengan adanya pembagian sumber daya, banyak proses dapat dirugikan karena terjadinya suatu kesalahan di dalam suatu program.

Tanpa adanya proteksi terhadap kesalahan ini, maka komputer harus mengeksekusi satu program pada satu saat saja atau semua output harus diduga. Suatu sistem

operasi yang dirancang dengan baik harus dapat menjamin bahwa program yang berjalan dengan tidak benar, tidak dapat menyebabkan program yang lain dieksekusi secara tidak benar. Banyak kesalahan pemrograman dideteksi oleh hardware. Kesalahan ini secara normal ditangani oleh sistem operasi. Ketika kesalahan program muncul, sistem operasi harus menghentikan secara tidak normal program tersebut, memberikan pesan kesalahan yang tepat, dan membebaskan memori yang dipakai oleh program tersebut. Jika suatu program pemakai mengalami perulangan tidak terbatas (*looping* abadi), maka program tersebut tidak akan membebaskan kembali CPU dan mengembalikan ke sistem operasi. Hal ini akan mengganggu kinerja sistem secara keseluruhan. Untuk mengatasi hal tersebut, sistem operasi menggunakan *timer*. *Timer* melakukan *interrupt* setelah perioda waktu tertentu untuk menjamin kontrol sistem operasi. *Timer* diturunkan setiap clock. Ketika timer mencapai nol, sebuah *interrupt* terjadi. *Timer* biasanya digunakan untuk mengimplementasikan pembagian waktu. *Timer* dapat juga digunakan untuk menghitung waktu sekarang walaupun fungsinya saat ini sudah digantikan *Real Time Clock (RTC)*.

#### DAFTAR PUSTAKA

Binanto, I. 2005. Sistem Operasi, Penerbit Andi, Yogyakarta.

Crowley, Charles. 1997. *Operating System: A Design-Oriented Approach*. Irwin. USA.

Heriyanto, B. 2005. Sistem Operasi Edisi ke-2, Penerbit Informatika, Bandung.

Pangera, A.A. 2008. Sistem Operasi. Penerbit Andi, Yogyakarta.

Stallings, William. 2001. *Operating System: Internals and Design Principles*. Prentice Hall, Inc. New Jersey. USA.